

IMPLEMENTASI SISTEM NOTIFIKASI KEADAAN DARURAT BERBASIS APLIKASI *MOBILE WEB* DAN ARDUINO MEGA MENGUNAKAN LOGIKA *FUZZY*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Prayoga Febriandika
NIM: 145150301111071



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI SISTEM NOTIFIKASI KEADAAN DARURAT BERBASIS APLIKASI
MOBILE WEB DAN ARDUINO MEGA MENGGUNAKAN LOGIKA FUZZY

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Prayoga Febriandika
NIM: 145150301111071

Skripsi ini telah diuji dan dinyatakan lulus pada
02 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Wijaya Kurniawan, S.T, M.T
NIP. 19820125 201504 1 002

Agi Putra Kharisma, S.T, M.T
NIK. 201304 860430 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 02 Juli 2018



Prayoga Febriandika

NIM: 145150301111071

KATA PENGANTAR

Puji syukur alhamdulillah, penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmatnya sehingga penulis dapat menyelesaikan penulisan skripsi dengan judul “IMPLEMENTASI SISTEM NOTIFIKASI KEADAAN DARURAT BERBASIS APLIKASI *MOBILE WEB* DAN ARDUINO MEGA MENGGUNAKAN LOGIKA *FUZZY*” untuk memenuhi sebagai persyaratan dalam memperoleh gelar Sarjana Fakultas Ilmu Komputer.

Penulis menyadari banyak sekali kesulitan dan hambatan dalam menyelesaikan skripsi ini terutama dalam mendapatkan data dan mengolahnya, akan tetapi semua itu telah diselesaikan dengan baik dari dukungan dan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan banyak sekali terima kasih kepada semua pihak yang telah bersedia membantu demi kelancaran penyusunan skripsi ini, diantaranya :

1. Kedua orang tua Bapak Ketut Hari Handoko dan Ibu Eny Rokhayati serta keluarga besar penulis yang selalu memberikan doa dan dukungan motivasi agar penulis dapat menyelesaikan skripsi ini dengan lancar.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T Ph.D. Selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D Selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Heru Nurwasito, Ir., M.Kom. Selaku Wakil Dekan Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. Selaku Ketua Program Studi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Wijaya Kurniawan, S.T, M.T Selaku Dosen Pembimbing I yang telah memberikan banyak sekali ilmu, saran, dan motivasi kepada penulis.
7. Bapak Agi Putra Kharisma, S.T, M.T Selaku Dosen Pembimbing II yang telah memberikan banyak sekali ilmu, pengarahan, dan sara kepada penulis.
8. Mohammad Fajar Mustofa, S.E dan seluruh civitas akademik FILKOM Universitas Brawijaya yang telah memberikan banyak bantuan.
9. Seluruh Dosen yang telah memberikan banyak sekali ilmu pengetahuan, dan serta Karyawan Fakultas Ilmu Komputer atas segala bantuan yang bersifat administratif.
10. Handoko Ramadhan, S.T dan Dea Valentina yang telah bersedia menjadi pembimbing pribadi serta banyak sekali membantu, memberikan dukungan, dan semangat untuk menyelesaikan proses skripsi.
11. Dimas Vico Putra Handoko sebagai Kakak yang telah memberikan doa dan dukungannya dari awal hingga saat ini.

12. Teman penulis : Oktaviany Setyowati, Fauzi Rivani, Faris Adhnaufal, Saudi Amran, Bramantyo Ardi, Rizky Teguh, Syahriel D, M. Taufik, Haqqi R, M Naufal, Joniar Dimas, Rafi Fajar, serta rekan seperjuangan Teknik Komputer Angkatan 2014.

Penulis menyadari bahwa laporan skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis mengucapkan terimakasih.

Malang, 02 Juli 2018

Penulis
pfebriandika@gmail.com



ABSTRAK

Keadaan darurat bagi setiap individu manusia biasanya dinyatakan saat terjadinya keadaan normal yang mempunyai kecenderungan atau potensi tingkat membahayakan baik bagi keselamatan manusia, harta benda maupun lingkungan. Pada saat manusia mengalami keadaan darurat, sangat diperlukan penanganan yang cepat untuk mengatasi kejadian tersebut. Belum adanya sistem notifikasi yang berperan sebagai penghubung antara korban dengan orang lain, saat korban mengalami keadaan darurat. Dengan adanya alat ini keluarga terdekat akan mengetahui kondisi *user* saat terjadi keadaan darurat. *User* hanya harus menekan tombol *button* sebagai *trigger* dan kemudian berteriak melalui sensor suara atau menjatuhkan diri saat terjadi keadaan darurat agar sistem dapat mengirim notifikasi ke aplikasi *mobile web*, yang nantinya aplikasi *mobile web* digunakan sebagai monitoring *user* oleh keluarga terdekat, dan keluarga terdekat akan mengetahui posisi terakhir korban melalui modul GPS. Pada penelitian ini terdapat metode *fuzzy sugeno* untuk menentukan sebuah kesimpulan pada sistem yang berupa kondisi *user* (normal atau darurat). Prototype sistem ini dibangun dengan menggunakan beberapa modul dan sensor, yaitu modul GPS, Modul ESP8266, *Push Button*, Sensor Suara, Sensor *Gyroscope Accelerometer* MPU6050, dan Arduino Mega sebagai mikrokontroler. Berdasarkan hasil uji coba sistem dapat diketahui bahwa perhitungan logika *fuzzy* secara manual sama dengan output logika *fuzzy* pada sistem. tingkat akurasi pada sistem ini sebesar 100% pada akurasi data sensor, logika *fuzzy* dan sistem dapat mengirimkan notifikasi pada saat user mengalami keadaan darurat.

Kata Kunci: Keadaan Darurat, Arduino Mega, Aplikasi *Mobile Web*, Logika *Fuzzy*

ABSTRACT

Emergency situation for every human individual Usually a normal condition that has a tendency or potential that is good for humans, property and the environment. At a time when humans experience an emergency, it is very necessary rapid handling to cover the incident. There is no system as a notification between the victim with others, when the victim experiences an emergency. With this tool, the closest family will know the condition of the user in the event of an emergency. Users only have to press the button as a trigger and then pass the sound sensor or send an emergency signal for the system to send notification to the mobile web app, which then the mobile web app as monitoring user by the closest family, and the family will know the last position of the victim through the GPS module when the user. In this research, fuzzy sugeno method is used to determine a system under the condition that is user condition (normal or emergency). The system prototype is built using several modules and sensors, ie GPS module, ESP8266 Module, Push Button, Sound Sensor, Gyroscope Accelerometer MPU6050 Sensor, and Arduino Mega as microcontroller. Based on the test results of the system can be seen that the calculation of logic fuzzy manually the same with the output of fuzzy logic on the system. the level of accuracy on this system of 100% on the accuracy of the data sensor, fuzzy logic and systems can send a notification upon user experiencing emergencies.

Keywords: Emergency, Arduino Mega, Mobile Web Application, Fuzzy Logic

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori	6
2.2.1 Keadaan Darurat	6
2.2.2 Aplikasi <i>Mobile Web</i>	6
2.2.2.1 <i>Web Server</i>	7
2.2.2.2 XAMPP	7
2.2.3 Sistem <i>Embedded</i>	8
2.2.4 Arduino Mega 2560	9
2.2.5 Modul ESP8266	11
2.2.6 Modul GPS Unblox NEO6M.....	12
2.2.7 <i>Push Button</i>	13
2.2.8 Sensor.....	13
2.2.8.1 Sensor Gyroscope Accelerometer MPU-6050	14

2.2.8.2	Sensor Suara LM393	14
2.2.9	Logika <i>Fuzzy</i>	15
2.2.9.1	Alasan Penggunaan Logika Fuzzy	15
2.2.9.2	Metode Inferensi Fuzzy Sugeno	16
2.2.9.3	Metode Fuzzy Sugeno dalam Mengambil Keputusan.....	16
2.2.10	Metode MIN-MAX.....	17
BAB 3	METODE PENELITIAN.....	19
3.1	Metode Penelitian	19
3.1.1	Studi literatur	20
3.1.2	Analisis Kebutuhan.....	20
3.1.2.1	Kebutuhan Fungsional.....	21
3.1.2.2	Kebutuhan Non-Fungsional.....	21
3.1.2.3	Kebutuhan Perangkat Keras	22
3.1.2.4	Kebutuhan Perangkat Lunak	22
3.1.3	Perancangan Sistem.....	23
3.1.4	Implementasi	23
3.1.5	Pengujian dan Analisis	24
3.1.6	Kesimpulan dan Saran.....	24
BAB 4	REKAYASA KEBUTUHAN.....	25
4.1	Gambaran Umum Sistem.....	25
4.2	Analisis Kebutuhan.....	25
4.2.1	Kebutuhan Fungsional	25
4.2.2	Kebutuhan Non-Fungsional	27
4.2.2.1	Kebutuhan Perangkat Keras	28
4.2.2.2	Kebutuhan Perangkat Lunak	29
4.3	Batasan Desain Sistem	29
BAB 5	PERANCANGAN DAN IMPLEMENTASI	30
5.1	Perancangan Sistem.....	30
5.1.1	Perancangan Alat Mekanik, Peletakan dan Ukuran sistem	30
5.1.2	Perancangan Perangkat Keras	31
5.1.3	Perancangan Perangkat Lunak.....	33
5.1.3.1	Perancangan Aplikasi <i>Mobile Web</i>	35

5.1.3.2 Perancangan Fuzzy	35
5.2 Implementasi Sistem.....	44
5.2.1 Implementasi Alat Mekanik, Peletakan dan Ukuran Sistem.....	45
5.2.2 Implementasi Perangkat Keras	45
5.2.3 Implementasi Perangkat Lunak.....	47
5.2.3.1 Implementasi Aplikasi <i>Mobile Web</i>	47
5.2.3.2 Implementasi Fuzzy Sugeno Pada Alat.....	50
BAB 6 PENGUJIAN DAN ANALISIS	58
6.1 Pengujian Akuisisi Data Sensor <i>Gyroscope</i>	58
6.2 Pengujian Akuisisi Data Sensor Suara	60
6.3 Pengujian Akuisisi Data Push Button	62
6.4 Pengujian Proses Fuzzy	64
BAB 7 PENUTUP	68
7.1 Kesimpulan.....	68
7.2 Saran	69
DAFTAR PUSTAKA.....	70

DAFTAR GAMBAR

Gambar 2.1 XAMPP Control Panel.....	7
Gambar 2.2 Arduino Mega 2560.....	9
Gambar 2.3 Modul Wifi ESP8266.....	11
Gambar 2.4 Modul GPS Unblox NEO6M.....	12
Gambar 2.5 <i>Push Button</i>	13
Gambar 2.6 Sensor <i>Gyroscope Accelerometer</i> MPU6050	14
Gambar 2.7 Sensor Suara LM393.....	15
Gambar 2.8 Perbandingan Metode Tsukamoto, Mamdani dan Sugeno	16
Gambar 3.1 Diagram Alir Metode Penelitian.....	19
Gambar 3.2 Analisis Kebutuhan.....	20
Gambar 3.3 Analisis Kebutuhan <i>User</i>	21
Gambar 3.4 Diagram blok Perancangan Sistem	23
Gambar 5.1 Skema Perancangan Sistem	30
Gambar 5.2 Skema Perancangan Perangkat Keras.....	31
Gambar 5.3 Rangkaian <i>Debounce</i>	32
Gambar 5.4 Flowchart Perancangan Perangkat Lunak.....	34
Gambar 5.5 Flowchart Perancangan Kegunaan Utama Aplikasi <i>Mobile Web</i>	35
Gambar 5.6 Flowchart Proses Perancangan <i>Fuzzy</i>	36
Gambar 5.7 Flowchart Proses Fuzzifikasi.....	36
Gambar 5.8 <i>Membership Function</i> Suara	38
Gambar 5.9 <i>Membership Function Gyroscope Y</i>	40
Gambar 5.10 <i>Membership Function Gyroscope P</i>	40
Gambar 5.11 <i>Membership Function Gyroscope R</i>	40
Gambar 5.12 Flowchart Proses Inferensi.....	42
Gambar 5.13 Flowchart Proses Defuzzifikasi.....	44
Gambar 5.14 Implementasi <i>Prototype</i> Sistem Tampak Samping	45
Gambar 5.15 Implementasi Rangkaian Perangkat Keras Tampak Samping	46
Gambar 5.16 Implementasi Rangkaian Perangkat Keras Bagian Dalam	46
Gambar 5.17 Menu Aplikasi <i>Mobile Web</i>	47
Gambar 5.18 Aplikasi <i>Mobile Web</i> pada Menu <i>Inbox</i>	48
Gambar 5.19 Tampilan pada Google Map	49
Gambar 5.20 Aplikasi <i>Mobile Web</i> pada Menu <i>Contact Police Area</i>	49
Gambar 5.21 Aplikasi <i>Mobile Web</i> pada Tampilan Notifikasi	50
Gambar 5.22 Kode Program Fuzzifikasi Suara	51
Gambar 5.23 Kode Program Fuzzifikasi <i>Sensor Gyroscope</i>	53
Gambar 5.24 Kode Program Rule dan Inferensi	55
Gambar 5.25 Kode Program Defuzzifikasi	55
Gambar 5.26 Kode Program <i>Output</i> Sistem	57

Gambar 6.1 Pengujian dan Analisis	58
Gambar 6.2 Grafik <i>Output</i> Sistem.....	66



DAFTAR TABEL

Tabel 5.1 Koneksi Pin ke Mikrokontroler di Perancangan Perangkat Keras.....	32
Tabel 5.2 Rule Logika <i>Fuzzy</i>	41
Tabel 6.1 Hasil Pengujian Sensor <i>Gyroscope</i>	59
Tabel 6.2 Hasil Pengujian Sensor Suara	61
Tabel 6.3 Hasil Pengujian <i>Push Button</i>	63
Tabel 6.4 Hasil Pengujian <i>Fuzzy</i>	67



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Keadaan darurat biasanya dinyatakan saat terjadinya keadaan yang muncul seperti terjadinya bencana alam, terjadinya kebakaran ataupun saat dinyatakan terjadinya perang. Akan tetapi bagi setiap individu manusia keadaan darurat terjadi saat keadaan normal yang mempunyai kecenderungan atau potensi tingkat membahayakan baik bagi keselamatan manusia, harta benda maupun lingkungan (Zie Ahmadi, 2013). Saat manusia mengalami keadaan darurat sangatlah diperlukan sebuah alat khusus untuk pihak yang berwajib dan juga keluarga terdekat agar dapat langsung mengetahui kejadian tersebut supaya korban bisa segera ditangani.

Penggunaan telepon genggam atau *handphone* terkadang belum bisa dikatakan efektif saat terjadi situasi darurat, karena pemakaian *hanphone* sendiri saat dalam keadaan panik pengguna tidak menggunakannya. Dalam kondisi situasi yang darurat, *handphone* tidak mempunyai fitur yang dapat langsung mengirim notifikasi seperti mengirim pesan atau lokasi pengguna dengan otomatis saat keadaan darurat seperti saat terjatuh, terjadi perampokan, pemerkosaan, penculikan, dan pembegalan. Maka dari itu masyarakat membutuhkan alat otomatis yang bisa mengirim notifikasi kepada keluarga yang sangat efisien dan mudah digunakan saat mengalami keadaan darurat.

Untuk menambah fitur yang mempermudah pemakai agar dapat menjadi alat yang efisien, maka dibuatlah sebuah penelitian berdasarkan sistem *embedded*. Ilmu yang mempelajari sistem yang dimana menggunakan komputasi untuk melakukan fungsi tertentu, tertanam dalam perangkat dan lingkungan yang lebih besar, atau sistem komputer yang dirancang untuk melakukan satu atau beberapa fungsi khusus, dengan komputasi *real-time*, meliputi perangkat keras, perangkat lunak dan bagian-bagian mekanis. Penelitian ini berfokus pada kegunaan sistem dengan mengutamakan *output* notifikasi dari aplikasi *mobile web*.

Peneliti juga akan menggunakan metode dari logika *fuzzy* Sugeno. Peneliti menggunakan logika ini karena logika *fuzzy* memiliki presisi yang tinggi dan solusi yang akurat, serta *output* dari sistem berupa value kondisi *user* yang dimana data-data dari sensor *input* terdapat ketidaktepatan atau menyebar dan data bisa ditoleransi menggunakan logika *fuzzy*. Serta logika *fuzzy* mampu memodelkan fungsi nonlinier yang sangat kompleks, dan mempresentasikan ketidakjelasan, ketidakpastian, serta kurangnya informasi (Sudradjat, 2008).

Pada penelitian ini penulis ingin menambahkan fitur untuk mengetahui *user*, saat *user* mengalami keadaan darurat dengan menekan *button* selama 7 detik untuk memastikan apakah benar terjadi keadaan darurat (*button* sebagai *trigger*), dan menambah *input* sensor suara untuk *user* dengan cara berteriak menggunakan suara yang lantang di sensor suara, serta peneliti juga menambahkan sensor *Gyroscope Accelerometer* untuk mengetahui kondisi *user* saat dalam keadaan terjatuh melalui sudut γ , p , dan r . Setelah data *input* yang didapat akan diolah menggunakan logika *fuzzy* sugeno untuk menentukan hasil akhir *ouput*, kemudian alat akan mengirim posisi terakhir yang didapat modul GPS ke aplikasi *mobile web*. Aplikasi *mobile web* berguna untuk pihak keluarga untuk menerima sebuah notifikasi pesan berupa kondisi *emergency* atau tidak, dan mengetahui keberadaan terakhir *user* menggunakan modul ESP sebagai pengiriman data. Perancangan di penelitian ini menggunakan Arduino Mega sebagai mikrokontroler sistem.

1.2 Rumusan Masalah

Dengan permasalahan yang dihadapi, penulis membuat perumusan masalah sebagai berikut ini :

1. Bagaimana cara mengimplementasikan sebuah sistem notifikasi saat keadaan darurat ke aplikasi *mobile web*?
2. Bagaimana cara menampilkan posisi *user* pada sebuah map di aplikasi *mobile web* menggunakan Arduino Mega sebagai mikrokontroler saat keadaan darurat?
3. Bagaimana cara mengolah data sensor suara untuk mendapatkan level *volume* sedang ke level *volume* tinggi menggunakan logika *fuzzy*?
4. Bagaimana cara mengetahui sudut kemiringan manusia berdasarkan kondisi untuk sensor *Gyroscope Accelerometer* menggunakan logika *fuzzy*?

1.3 Tujuan Penelitian

Dengan dirumuskannya masalah, didapatkan tujuan yang terkait pada rumusan masalah sebagai berikut ini :

1. Dapat mengetahui bagaimana cara mengimplementasi sistem untuk mengirim notifikasi ke aplikasi *mobile web*.
2. Dapat menampilkan posisi pengguna di aplikasi *mobile web* saat keadaan darurat.
3. Dapat mengolah data *input* sensor suara dan mendapatkan nilai level *volume* suara manusia menggunakan logika *fuzzy*.
4. Dapat mengetahui sudut kemiringan normal manusia maupun kemiringan saat terjatuh dan menerapkannya di logika *fuzzy*.

1.4 Manfaat Penelitian

Dari fungsi sistem yang akan diteliti penulis, didapatkan manfaat penelitian sebagai berikut :

1. Kebanggaan bagi diri sendiri ketika menciptakan sebuah sistem yang dapat membantu orang lain maupun keluarga.
2. Dapat mengurangi jumlah angka kriminalitas akibat kurang cepatnya pertolongan saat terjadi keadaan darurat.
3. Dapat memonitoring keluarga saat ada yang ingin berpergian jauh atau keluar pada malam hari.
4. Dapat membantu pihak berwajib untuk melaksanakan tugasnya dengan cepat.

1.5 Batasan Masalah

Berdasarkan rumusan masalah yang telah disampaikan di atas, maka batasan masalah pada penelitian ini adalah:

1. Sistem diperuntukan untuk kondisi saat *user* mengalami keadaan darurat. dan langsung mengirim notifikasi pesan ke aplikasi *mobile web* agar mendapatkan pertolongan cepat.
2. Keadaan darurat merupakan kejadian seperti penculikan, pemerkosaan, perampokan, kehilangan seseorang dan kecelakaan.
3. Implementasi sistem ini menggunakan mikrokontroler Arduino Mega 2560.
4. Pengoperasian utama pada sistem ini menggunakan sensor *Gyroscope Accelerometer*, sensor suara sebagai *input* data, dan 1 *button* sederhana sebagai *trigger*.
5. Sistem pada aplikasi *mobile web* berguna untuk menerima notifikasi kondisi dan posisi terakhir saat *user* mengalami keadaan darurat.

1.6 Sistematika pembahasan

Penulisan skematik pembahasan dan penyusunan laporan penelitian dapat diuraikan sebagai berikut ini :

BAB I : PENDAHULUAN

Di bab I ini Pendahuluan akan dijelaskan tentang yang berhubungan dengan latar belakang, tujuan, rumusan masalah dan juga manfaat dari penulisan skripsi. Bab ini akan menjadi dasar dari keseluruhan perancangan sistem.

BAB II : LANDASAN TEORI

Pada Bab II ini akan dijelaskan tentang penelitian yang terdapat referensi terkait dan memiliki bahasan dan tujuan atau berita melalui lembaga resmi yang memungkinkan agar bisa dipertanggung jawabkan, berupa pengertian dan definisi. Bab ini juga menjelaskan alasan penulis menggunakan perangkat-perangkat sensor dan informasi yang nantinya bakal saling berkaitan dengan batasan masalah yang penulis berikan.

BAB III : METODE PENELITIAN

Membahas metode yang digunakan didalam penelitian, terdiri dari studi literatur, perancangan, implementasi, pengujian, dan analisis, serta adanya pengambilan kesimpulan dan saran.

BAB IV : ANALISIS KEBUTUHAN

Pada bab IV akan menguraikan seluruh kebutuhan yang bertujuan agar sistem dapat bekerja sesuai dengan tujuan, yang terdiri dari kebutuhan *user*, kebutuhan fungsional sistem, dan kebutuhan lainnya yang diperlukan sistem ini.

BAB V : PERANCANGAN DAN IMPLEMENTASI

Bab V membahas mengenai penjelasan Perancangan sistem yang dibuat dan imlementasi system yang dibuat yang berupa perangkat keras dan batasan implementasi.

BAB VI PENGUJIAN DAN ANALISA

Pada bab ini menjelaskan tentang hasil pengujian serta analisis dari sistem notifikasi keadaan darurat.

BAB VII : PENUTUP

Bab terakhir ini akan memuat tentang kesimpulan yang berasal dari pembahasan rumusan pada masalah yang ada dan berdasarkan dari sebuah analisis dan .pengujian yang .sudah dilakukan dalam .proses penelitian, serta terdapat adanya saran sebagai pengembangan kedepan yang lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Berikut beberapa adanya penelitian yang terkait sistem notifikasi saat keadaan darurat. Beberapa penelitian terkait belum adanya implementasi alat langsung yang dapat mengirim notifikasi saat terjadinya keadaan darurat dan hanya terdapat dalam aplikasi maupun *web* sebagai *input* serta tidak adanya metode sebagai pengambilan keputusan sistem.

Pada penelitian yang mempresentasikan dengan judul “Rancang Bangun *Automatic Emergency System* Berbasis *Mobile*”. Penelitian ini berfokus pada sistem darurat berupa aplikasi *mobile* untuk memberikan pelayanan secara cepat kepada masyarakat. Bertujuan menyelesaikan permasalahan yang masih lambatnya respon sistem pelayanan publik. Sistem *automatic emergency* ini dikembangkan menggunakan metode *system development life cycle (SDLC)*. Sedangkan implementasi sistem dibuat menggunakan bahasa pemrograman *PHP*, sistem operasi android, dan *MYSQL-Server* sebagai basis data (Henderi, Nanda Dian, & Didik, 2016).

Pada penelitian yang berjudul “Rancang Bangun Sistem Informasi Tanggap Darurat Divisi Pusat Pengaduan Berbasis Website di Kota Palembang” yang bertujuan untuk mengatasi kelambatan memberikan informasi kejadian darurat kepada petugas pengaduan masyarakat, serta penelitian ini berfokus pada sebuah website sistem. Proses pada pengembangan sistem menggunakan metodologi *Rational Unified Process (RUP)* yang meliputi fase *inception*, fase *elaboration*, fase *construction*, dan fase *transition* yang disetiap fase dapat melaksanakan secara berulang-ulang sampai mendapatkan hasil yang diinginkan. Sedangkan untuk perangkat lunak, sistem ini menggunakan *Xampp*, *PHP*, dan *MySQL*. Hasil *output* sistem ini yaitu menyediakan informasi mengenai letak fasilitas yang dapat digunakan di pemanggilan darurat dan memberikan informasi pemanggilan darurat kepada petugas (Jeffrey, Heryanto, & Iis, 2016).

Pada penelitian lainnya yang berjudul “*Prototype Panic Button* Dengan Notifikasi SMS dan Peta Digital” bertujuan untuk membantu dalam pemanggilan pihak kepolisian, dinas pemadam kebakaran, dan rumah sakit agar dapat mendapatkan penanganan lebih cepat. Dengan menekan *button* sebagai acuan untuk mengirimkan notifikasi berupa SMS dan dapat langsung mengetahui lokasi kejadian tindak kriminal berada atau keadaan darurat terjadi. Dengan menggunakan mikrokontroler berupa *ATMega328* atau *Arduino Nano*, serta menggunakan *push button* sebagai *input*. Kemudian mengirim melalui *web server* serta menyambungkan telepon ke pihak berwajib (Rizki & Sutanto, 2017)

Berdasarkan pada tiga referensi penelitian yang sudah dilakukan diatas, terdapat kemiripan pada objek penelitian yaitu berupa ide yaitu notifikasi saat keadaan darurat atau penanganan cepat disaat keadaan darurat serta aplikasi *mobile* yang digunakan hingga cara kerja sistem yang digunakan. Sehingga peneliti memilih tiga referensi tersebut sebagai referensi penelitian. Sensor yang akan digunakan pada penelitian penulis yaitu sensor *Gyroscope* dan sensor suara *microphone* serta *push button* sebagai *trigger*. Untuk penerima data *output* penulis menggunakan aplikasi *mobile web* serta Arduino Mega 2560 sebagai mikrokontroler.

2.2 Dasar Teori

2.2.1 Keadaan Darurat

Keadaan darurat merupakan suatu kejadian, kondisi atau keadaan yang tidak normal dan tidak direncanakan, dimana keadaan tersebut terjadi secara tiba-tiba yang mempunyai kecenderungan potensi membahayakan bagi keselamatan manusia, lingkungan maupun harta benda. Maka dalam situasi atau kejadian keadaan darurat haruslah dilakukan penanganan secara cepat agar tidak mengakibatkan adanya korban, kerusakan atau kehilangan.

2.2.2 Aplikasi *Mobile Web*

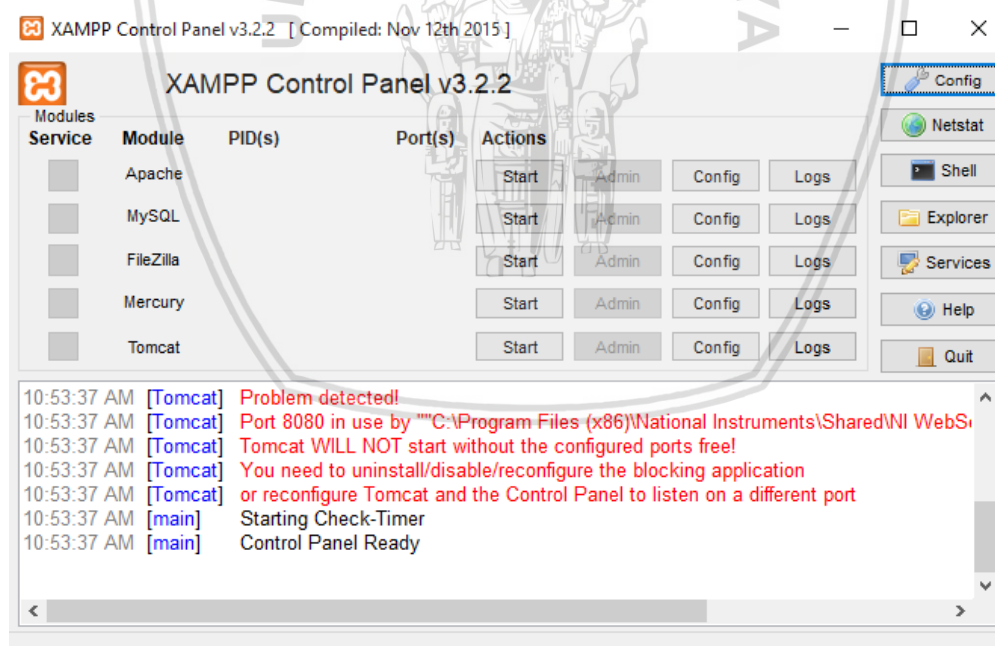
Mobile web merupakan sebuah halaman *HTML* yang berbasis *browser* kemudian diakses menggunakan perangkat portable melalui jaringan telekomunikasi seluler (Wifi, 3G/4G) yang dirancang untuk menampilkan teks konten, gambar, video, dan data. Prinsip dari *mobile web* yaitu dapat beroperasi pada lintas platform dalam sekali pengembangan. *Mobile web* juga dapat didistribusikan tanpa harus mendapatkan approval dari pihak tertentu, akan tetapi *mobile web* harus memiliki dua buah domain sebagai kestabilan sistem dan juga keamanan sistem. *Mobile web* dapat diubah menjadi aplikasi *mobile apps*, berbeda dengan aplikasi *mobile apps* yang tidak bisa diubah menjadi *mobile web*.

2.2.2.1 Web Server

Web server merupakan salah satu kebutuhan yang digunakan oleh *user* untuk website yang mempunyai kapasitas penyimpanan yang besar dan juga akses yang cepat untuk trafik yang besar dalam mencegah terjadinya *down* pada suatu website atau aplikasi. Web server atau server *web* merupakan perangkat lunak (*software*) dalam server yang berfungsi untuk menerima permintaan (*request*) berupa halaman *web* melalui protokol HTTP atau HTTPS dari client yang lebih dikenal dengan nama browser, kemudian mengirimkan kembali atau merespon hasil permintaan tersebut ke dalam bentuk halaman-halaman *web* yang pada umumnya berupa dokumen HTML atau PHP.

2.2.2.2 XAMPP

XAMPP merupakan suatu alat (*tool*) yang menyediakan beberapa perangkat lunak ke dalam satu buah paket. Contoh *software* yang berada di dalam XAMPP diantaranya adalah Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FileZilla, MySQL dan lainnya. Setelah menginstall XAMPP, perangkat lunak yang telah disebutkan di atas akan terkonfigurasi secara otomatis. Tampilan antarmuka dari XAMPP Control Panel dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 XAMPP Control Panel

Di dalam Gambar 2.1 terlihat beberapa fungsi atau fitur utama dari XAMPP Control Panel, yaitu Apache, MySQL, FileZilla, Mercury dan Tomcat. Fitur-fitur tersebut memiliki fungsi pembantu untuk membuat sebuah webserver.

Apache diartikan sebagai suatu *web server* yang kompak, modular, mengikuti standar protokol HTTP. *Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. *Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

PHP merupakan bahasa pemrograman *web* yang bersifat *server-side scripting* dimana *request* dari user dijalankan dengan cara menjalankan script langsung pada *web server (Apache)*.

MySQL adalah sebuah sistem manajemen database yang bersifat *open source* yang digunakan untuk membuat dan mengelola database beserta isinya seperti menambah, mengubah dan menghapus isi data yang berada di dalam database. MySQL memiliki sistem manajemen database yang bersifat *relational* yang berarti data-data yang dikelola dalam database diletakkan pada tabel-tabel terpisah namun terhubung satu sama lain sehingga proses manipulasi data lebih cepat.

2.2.3 Sistem *Embedded*

Sistem *embedded* adalah sebuah *computing device* yang didesain dengan tujuan tertentu dengan spesifik untuk melakukan fungsi tertentu sesuai yang diinginkan pembuat. Sistem *embedded* sendiri terdiri dari perangkat keras dan juga perangkat lunak. Dari perangkat kerasnya yaitu meliputi mikrokontroler dengan penambahan memori eksternal, *input* dan *output*, dan berbagai komponen lainnya seperti sensor, LED, dan akuntor lainnya. Sedangkan dari perangkat lunak *embedded* yaitu penggerak pada sistem *embedded*. Sebagian penggerak pada sistem *embedded* itu *real time* yang memiliki program aplikasi yang spesifik yang didukung oleh *Real Time Operating (RTOS)*. Yang terpenting disini yaitu sekali program dimasukkan kedalam perangkat keras maka tidak akan pernah berubah kecuali di program ulang. Terdapat juga kategori dalam sistem *embedded* yaitu sistem *embedded* berdiri sendiri (*Stand Alone*), Sistem *Embedded Real-Time*, Sistem *Embedded Hard Real-Time*, Sistem *Embedded Soft Real-Time*, dan *Networked Embedded Systems*.

2.2.4 Arduino Mega 2560

Arduino Mega merupakan mikrokontroler yang berbasis chip ATmega328P dengan bentuk yang lebih besar daripada Arduino Nano maupun Arduino Uno. Board ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin analog *input*, 4 pin UART (*serial port hardware*). Arduino Mega 2560 dilengkapi dengan sebuah oscillator 16 Mhz, sebuah port USB, power jack DC, ICSP header, dan tombol reset. Board ini sudah sangat lengkap, sudah memiliki segala sesuatu yang dibutuhkan untuk sebuah mikrokontroler. Dengan penggunaan yang cukup sederhana, tinggal menghubungkan power dari USB ke PC atau melalui adaptor AC/DC ke jack DC (Arduino, 2016). Spesifikasi yang dimiliki Arduino Mega adalah sebagai berikut serta contoh Arduino Mega 2560 pada Gambar 2.2.

1. Mikrokontroler : Arduino Mega
2. Operating Voltage : 5V
3. Flash Memory : 256 KB *of which 8 KB used by bootloader*
4. SRAM : 8 KB
5. Clock Speed : 16 MHz
6. Analog I/O Pin : 16 buah
7. EEPROM : 4 KB
8. DC Current per I/O : 20 mA
9. *Input Voltage* : 7-12 V
10. Digital I/O Pin : 54 buah
11. PWM *Output* : 6
12. *PCB Size* : 101,5 mm x 53,4 mm
13. *Weight* : 37 g



Gambar 2.2 Arduino Mega 2560

Sumber : (www.technobotsonline.com, 2002)

Dari Gambar 2.2 diatas dapat dijelaskan secara fungsional konfigurasi pin pada Arduino Mega sebagai berikut:

1. Vin : *Input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal
2. GND : Pin ground
3. 5V : Sebuah pin yang mengeluarkan tegangan *regulator* 5 Volt, dari pin ini tegangan sudah diatur (*regulator*) dari *regulator* yang tersedia (*built-in*) pada papan.
4. 3V3 : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh *regulator* yang terdapat pada papan (*on-board*).
5. IOREF : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler.
6. Eksternal Interrupt : Pin ini dapat dikonfigurasi untuk memicu sebuah *interupsi* pada nilai yang rendah, meningkat atau menurun, atau perubah nilai. PWM : 3, 5, 6, 9, 10 dan 11. Menyediakan *output* PWM 8 bit dengan fungsi *analogWrite()*.
7. SPI : 10 (SS), 11 (MIA), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI walaupun disediakan oleh *hardware*, saat ini tidak termasuk dalam Bahasa Arduino.
8. LED : 13, ada LED *built-in* yang terhubung ke pin digital 13 apabila pin bernilai High, LED akan menyala dan apabila LED bernilai Low, LED akan mati.
9. I2C : A4 (SDA) dan A5 (SCL), merupakan komunikasi yang Mendukung I2C (TWI) dengan menggunakan *wire library*.
10. AREF : Tegangan referensi untuk *input* analog. Digunakan dengan *analogReference()*.
11. Reset : pin untuk mereset mikrokontroler.

2.2.5 Modul ESP8266

Modul WiFi ESP8266 adalah *SoC (System on Chip)* mandiri yang dilengkapi tumpukan protokol TCP/IP terintegrasi yang dapat memberi akses mikrokontroler ke jaringan WiFi. ESP8266 mampu meng-host aplikasi atau membongkar semua fungsi jaringan Wi-Fi dari prosesor aplikasi lain. Setiap modul ESP8266 hadir diprogram dengan perintah AT, yang artinya bisa menghubungkannya dengan perangkat Arduino Anda dan mendapatkan sebanyak mungkin kemampuan WiFi sebagai pilihan WiFi Shield. Sehingga bisa melakukan programming langsung ke ESP8266 tanpa memerlukan mikrokontroler tambahan.



Gambar 2.3 Modul Wifi ESP8266

Sumber : (www.sinauarduino.com, 2016)

Modul ini memiliki kemampuan pemrosesan dan penyimpanan on-board yang cukup kuat yang memungkinkannya untuk diintegrasikan dengan sensor dan perangkat spesifik aplikasi lainnya melalui GPIO-nya dengan perkembangan minimal di atas dan pemuatan minimal selama *runtime*. Tingkat integrasi on-chipnya yang tinggi memungkinkan sirkuit eksternal minimal, termasuk modul front-end, dirancang untuk menempati area PCB minimal. Kelebihan lainnya, ESP8266 ini dapat menjalankan peran sebagai adhoc akses poin maupun klien sekaligus. ESP8266 mendukung APSD untuk aplikasi VoIP dan antarmuka koisten Bluetooth, ini berisi RF yang dikalibrasi sendiri yang memungkinkannya bekerja dalam semua kondisi operasi, dan tidak memerlukan bagian RF eksternal. Dan modul ini biasanya menjadi solusi IO (*Internet of Things*). Contoh Modul ESP8266 pada Gambar 2.3.

2.2.6 Modul GPS Unblox NEO6M

Global Positioning System (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Spesifikasi yang dimiliki Modul GPS unblox NEO6M adalah sebagai berikut serta contoh Modul GPS unblox NEO6M pada Gambar 2.4.

1. Receiver type : 50 channels, GPS L1 Freq, C/A code
2. SBAS : WAAS, EGNOS, MSAS
3. First Fix time : 27s (cold/warm start), 1s (hot start), <3s (aided start)
4. Sensitivity : -161 dBm (Tracking and navigation), -160 dBm (Reacquisition), -147 dBm (Cold start), -156 dBm (Hot start)
5. Max update rate : 1 Hz
6. Timepulse freq range : 0.25 - 1 Hz
7. Horizontal accuracy : 2.5m (GPS), 2.0m (SBAS)
8. Timepulse signal accuracy : 30 ns (RMS), <60 ns (99%), 21 ns (Granularity), 15 ns (Compensated)
9. Velocity accuracy : 0.1m/s
10. Heading accuracy : 0.5 degree
11. Operational limits : 4g (dynamics), 50.000m (altitude), 500 m/s (velocity)
12. Input voltage : 3.3 V - 5 V
13. Logic level : 3.6 V



Gambar 2.4 Modul GPS Unblox NEO6M

Sumber : (<http://ecadio.com>, 2018)

2.2.7 Push Button

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal. Contoh *push button* seperti Gambar 2.5.



Gambar 2.5 *Push Button*

Sumber : (www.sparkfun.com, 2003)

2.2.8 Sensor

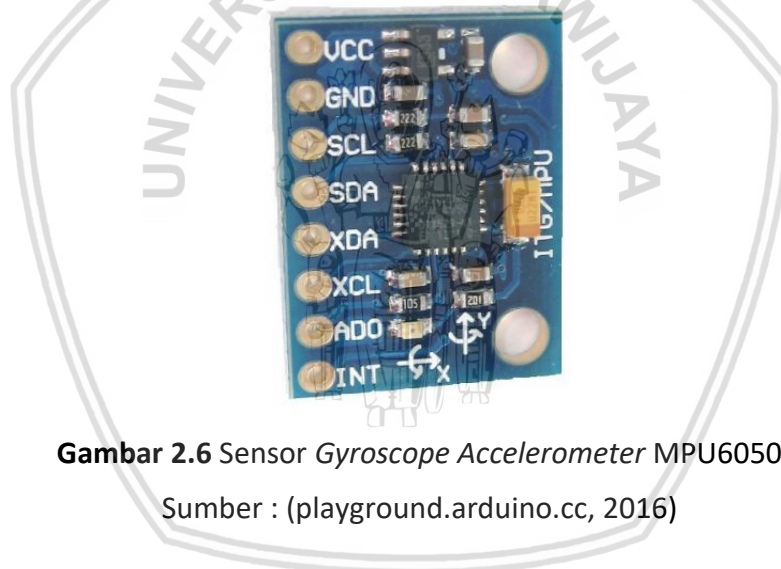
Sensor merupakan suatu jenis *tranduser*, fungsi sensor untuk pengubah besaran magnetis, mekanis, sinar, panas, kimia ke besaran listrik yang berupa tegangan, arus listrik, dan resistansi. Sensor sendiri sering digunakan juga untuk mendeteksi sesuatu pengukuran dan pengendalian.

Terdapat banyak kegunaan fungsi sensor, salah satunya sensor untuk keamanan yaitu perlindungan terhadap object tertentu dari kerusakan yang ditimbulkannya sendiri, serta keamanan untuk peralatan, komponen, dan perlindungan makhluk hidup disekitar lingkungan dimana object tersebut digunakan.

2.2.8.1 Sensor Gyroscope Accelerometer MPU-6050

Sensor *Gyroscope Accelerometer* MPU-6050 merupakan modul sensor yang berisi MEMS (*Microelectromechanical systems*) untuk *Accelerometer* (sensor percepatan) dan *Gyroscope* (pengatur keseimbangan) dalam satu chip yang super kecil. Kemampuan sensor ini sangatlah akurat, karena berisi *hardware* khusus untuk konversi analog ke digital selebar 16-bit untuk masing-masing chanelnya. Selain itu sensor ini sudah memiliki *Digital Motion Processor* (DMP) yang berfungsi mengolah data mentah dari masing-masing sensor dan berfungsi meminimalisasi *error* yang dihasilkan.

Sensor *Gyroscope* yang terdapat pada MPU-6050 merupakan perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut. *Gyroscope* pada MPU-6050 dapat membaca kecepatan sudut yang dinamis serta *output* yang dihasilkan berupa kecepatan sudut dari arah 3 sumbu yang didapatkan secara bersamaan, yaitu sumbu Y (atas-bawah), P (kanan-kiri), dan R (depan-belakang). Sehingga membuat alat ini mampu bekerja secara cepat dan presisi. Contoh sensor *Gyroscope* seperti Gambar 2.6.

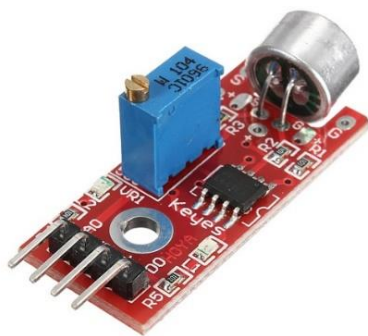


Gambar 2.6 Sensor *Gyroscope Accelerometer* MPU6050

Sumber : (playground.arduino.cc, 2016)

2.2.8.2 Sensor Suara LM393

Sensor suara ini adalah tambahan sensor untuk keamanan yang berbasis Arduino. Jadi selain menggunakan PIR untuk deteksi keberadaan seseorang juga bisa ditambah dengan sensor suara sehingga jika tidak lewat di depan PIR tetapi jika terdengar suara maka akan mentrigger sistem keamanan baik berupa alarm maupun mengirim SMS notifikasi. Sensor ini memiliki 3 pin yang dimana *output*nya adalah *output* analog. Selain itu, sensor ini memiliki fasilitas untuk mengatur sensitivitas suara yang ditangkap. Contoh Sensor Suara seperti Gambar 2.7.



Gambar 2.7 Sensor Suara LM393

Sumber : (www.okazii.ro, 2017)

2.2.9 Logika Fuzzy

Teorema *Fuzzy* merupakan pengingkaran dari logika boolean yang dapat mengenalkan konsep *kebenaran sebagian*. Di mana logika ini menyatakan bahwa dalam segala hal yang bisa diekspresikan dalam istilahnya yaitu binary 0 atau 1, ya atau tidak, hitam atau putih.

Logika *Fuzzy* ini juga akan memungkinkan sebuah nilai yang keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih dalam bentuk linguistik, konsepnya sendiri tidak begitu pasti, seperti contohnya "sedikit", "sedang", dan atau "sangat". Dia akan berhubungan langsung dengan set *fuzzy* dan juga teori kemungkinan. Logika ini diperkenalkan Dr. Lotfi Zadeh dari Universitas California di Berkeley pada tahun 1965 (Sumber: Gembong.lecture.ub.ac.id).

2.2.9.1 Alasan Penggunaan Logika Fuzzy

1. Logika *Fuzzy* ini sangatlah fleksibel untuk digunakan.
2. Konsep dari logika *fuzzy* sangatlah mudah untuk dimengerti. Konsep matematisnya yang begitu sederhana dan juga mudah untuk dimengerti.
3. Logika *Fuzzy* sendiri sangatlah memiliki toleransi.
4. Logika *fuzzy* begitu mampu memodelkan sebagai fungsi dalam nonlinear yang kompleks.
5. Logika *fuzzy* memudahkan untuk membangun dan juga mengaplikasikan sebagai pengalaman-pengalaman oleh para pakar tertentu secara langsung dan tanpa harus melalui banyak proses pelatihan.

2.2.9.2 Metode Inferensi Fuzzy Sugeno

Inferensi *fuzzy* menggunakan metode sugeno, pada metode ini mempunyai ciri-ciri berupa konsekuen dan bukan merupakan himpunan *fuzzy*, akan tetapi merupakan sebuah persamaan linier dengan variabel-variabel yang cocok dengan variabel *input*nya. Metode ini pertama kali diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Terdapat 2 model untuk menggunakan metode Sugeno yaitu model Sugeno Orde-0 dan Model Sugeno Orde-1 (Naba, 2009).

Ouput pada rule/aturan bukan dalam bentuk fungsi keanggotaan, akan tetapi suatu bilangan yang dapat berubah secara linier terhadap variabel *input*, yaitu mengikuti suatu persamaan bidang $z=av+bw+c$. Jika $b=0$ maka sistem inferensi beroder satu(1) yang dimana keluarannya mengikuti persamaan garis. Jika suatu persamaan bidang $z=av+c$, jika $a=b=0$ maka sistem inferensi dikatakan berorder nol karena keluarannya berupa konstanta, yaitu $z=c$.

Berikut merupakan perbandingan beberapa macam metode *fuzzy* yang dijelaskan pada Gambar 2.8.

Metode Fuzzy	Input	Output	Defuzzifikasi	Penggunaan
Tsukamoto	Himpunan Fuzzy	Himpunan Fuzzy	Weigthed Average	Humanis Control
Mamdani	Himpunan Fuzzy	Himpunan Fuzzy	CoG, Lom, Som, Mom, Bisector	Humanis
Sugeno	Himpunan Fuzzy	Orde 0 atau 1	Weigthed Average	Control

Gambar 2.8 Perbandingan Metode Tsukamoto, Mamdani dan Sugeno

Sumber : (departemen ilmu computer ITB,2012)

2.2.9.3 Metode Fuzzy Sugeno dalam Mengambil Keputusan

Penalaran pada metode ini memiliki banyak persamaan dengan penalaran Mamdani, yang membedakan adalah *output*/keluaran pada sistem bukan merupakan himpunan *fuzzy*, tetapi nilai konstanta atau persamaan linier.

2.2.10 Metode MIN-MAX

Fungsi implikasi MAX-MIN dan MAX-PROD merupakan metode yang paling umum untuk digunakan karena mudah untuk diimplementasikan dan bila diagregasikan dengan fungsi lainnya akan diperoleh bentuk yang mudah di *defuzzifikasi*. Pada metode MAX(Maximum) menggunakan penyelesaian himpunan *fuzzy* yang didapatkan dengan cara mengambil nilai terbesar dari rule/aturan, lalu menggunakannya untuk dimodifikasi daerah *fuzzy* dan menerapkan kedalam *output* dengan menggunakan operator OR(Union). Apabila semua proposisi sudah dilakukan evaluasi, maka *output* sistem akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi pada tiap-tiap proposisi. Metode MIN-MAX menggunakan konstanta maupun fungsi pada variabel input :

Jika a adalah A_i dan b adalah B_i maka c adalah $C_i = f(a,b)$

Dengan a, b, dan c yang merupakan sebuah variabel linguistik; A_i dan B_i himpunan *fuzzy* ke- i untuk a dan b, dan $f(a,b)$ merupakan fungsi matematik. Untuk mendapat *output*(hasil), maka terdapat 4 langkah / tahapan sebagai berikut:

1. Pembentukan himpunan *fuzzy*

Langkah pertama dalam *fuzzy* yaitu membentuk himpunan keanggotaan pada masing-masing *input fuzzy*.

2. Aplikasi fungsi implikasi

Membuat aturan-aturan berupa implikasi dari *fuzzy* yang menyatakan relasi antara variabel *input* dan *output* pada *fuzzy*. Pada sistem inferensi *fuzzy* sugeno, fungsi implikasi yang digunakan adalah Min. Bentuknya adalah sebagai berikut:

Jika a adalah A_i dan b adalah B_i maka c adalah $C_i = f(a,b)$

3. Inferensi

Inferensi didapatkan dari kumpulan dan korelasi antara rule/aturan. Metode yang digunakan untuk inferensi *fuzzy* adalah metode MIN-MAX. Himpunan *fuzzy* didapatkan dengan mengambil nilai minimum aturan, lalu menggunakan nilai tersebut untuk memodifikasi daerah *fuzzy* yang kemudian menerapkannya ke dalam *output* dengan menggunakan operator AND. Apabila semua proposisi sudah dilakukan evaluasi, maka pada *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi pada setiap proposisi. Secara umum dapat dituliskan pada persamaan:

$$\mu(x_i) = \max(\mu_{sf}(x_i), \mu_{kf}(x_i)) \dots \dots \dots (2.1)$$

dengan :

- $\mu_{sf}(x_i)$ = merupakan nilai keanggotaan solusi pada *fuzzy* sampai rule/aturan ke-i
- $\mu_{kf}(x_i)$ = merupakan nilai keanggotaan konsekuen *fuzzy* sampai rule/aturan ke-i

4. Defuzzifikasi

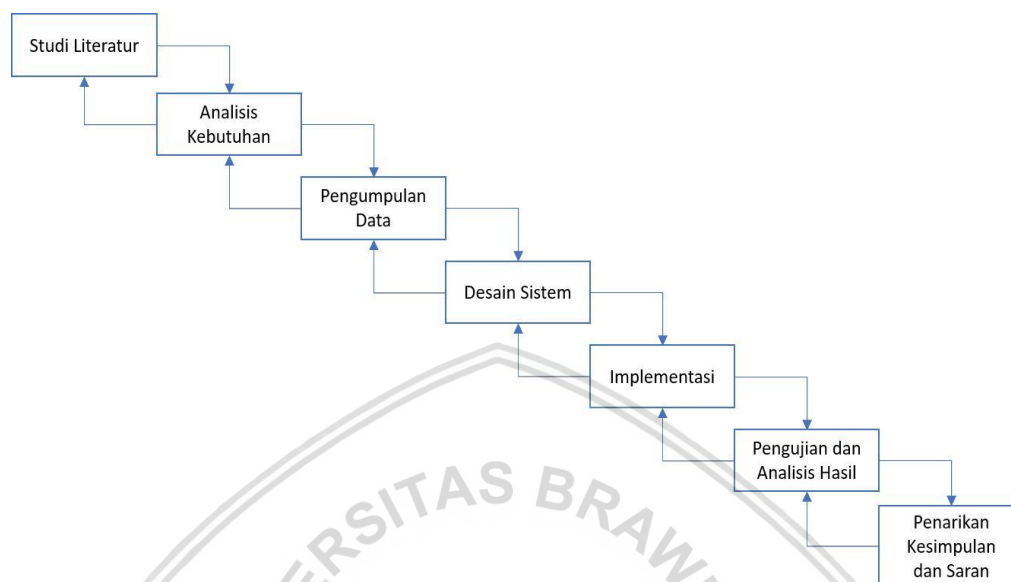
Input pada proses penegasan merupakan suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan *fuzzy*, sedangkan *output* yang dihasilkan berupa suatu bilangan real yang tegas. Apabila diberikan suatu himpunan *fuzzy* pada range tertentu, maka bisa diambil suatu nilai tegas tertentu sebagai *output*. Jika komposisi aturan menggunakan metode sugeno, maka pada proses defuzzifikasi dilakukan dengan cara mencari nilai rata-rata terpusatnya. Seperti pada persamaan berikut :

$$C = \max(a,b) \dots \dots \dots (2.2)$$



BAB 3 METODE PENELITIAN

3.1 Metode Penelitian



Gambar 3.1 Diagram Alir Metode Penelitian

Bab ini menjelaskan metode yang nantinya digunakan dalam melakukan penelitian tentang Implementasi Sistem Notifikasi Keadaan Darurat Berbasis Aplikasi *Mobile Web* dan Arduino Mega Menggunakan Logika *Fuzzy*. Langkah pertama yang dilakukan adalah mengumpulkan teori-teori sebagai pendukung penelitian dan akan dikemas dalam studi literatur. Kemudian akan dilanjutkan dengan proses analisis kebutuhan. Setelah tahap analisis kebutuhan, proses selanjutnya yaitu melakukan proses pengumpulan data yang dilanjutkan dengan proses implementasi *hardware* dan serta implementasi *software* sesuai dengan perancangan. Tahap berikutnya dilakukan pengujian dan analisis pada rancangan yang telah dibangun untuk menjadi tolak ukur kinerja sistem agar sesuai dengan apa yang sudah direncanakan. Terdapat juga kesimpulan dan saran disertakan sebagai catatan atas rancangan dan kemungkinan arah pengembangan selanjutnya.

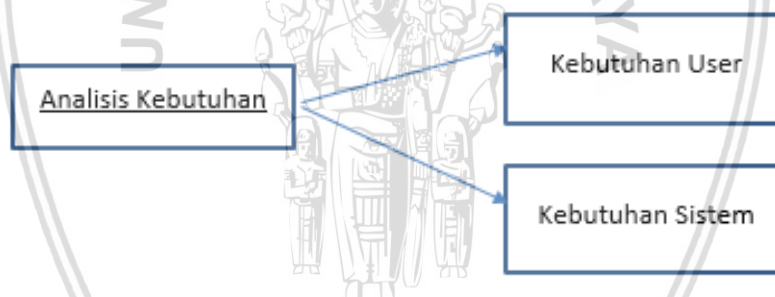
3.1.1 Studi literatur

Studi Literatur merupakan salah satu teknik yang digunakan dalam melaksanakan sebuah penelitian. Studi Literatur yang berfungsi agar menyelesaikan persoalan melalui penelusuran sumber-sumber yang tulisannya pernah dibuat sebelumnya. Dan bisa juga dikatakan Studi ini sangat familier atau bisa disebut studi pustaka. Studi literatur disini dilakukan melalui cara pengumpulan teori dan pustaka yang berkaitan dalam suatu penelitian sebagai berikut ini :

1. Mekanisme penghubungan sensor dengan mikrokontroler.
2. Pemograman di Arduino MEGA 2560.
3. Mikrokontroler Arduino MEGA 2560.

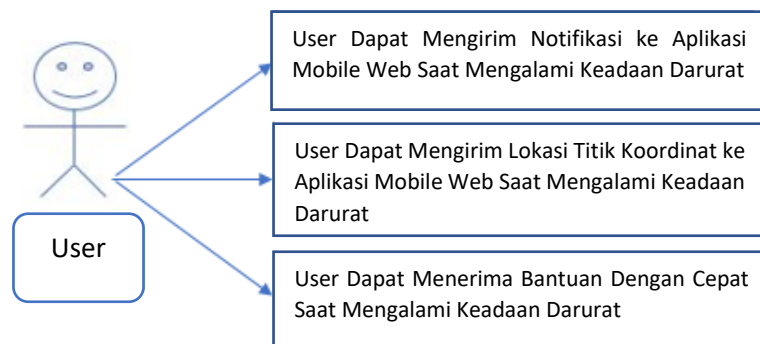
3.1.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengidentifikasi perangkat yang akan digunakan dalam membangun dan mengimplementasikan sistem. Perangkat yang digunakan harus sesuai dengan peran yang akan dijalankan dalam kerja sistem sehingga perancangan akan lebih mudah. Analisis kebutuhan terdiri dari dua yakni kebutuhan sistem dan kebutuhan *user*.



Gambar 3.2 Analisis Kebutuhan

Didalam analisis kebutuhan *user* hanya menjelaskan kebutuhan secara fungsional. Terdapat Beberapa poin yang menjelaskan hal agar *user* dapat melakukan kepada sistem yang akan menjadi kebutuhan *user* terhadap sistem yang dibangun. Kebutuhan *user* dapat di defisinikan seperti Gambar 3.3.



Gambar 3.3 Analisis Kebutuhan *User*

Didalam Tahap analisis kebutuhan sistem, terdapat 2 kebutuhan yaitu fungsional dan non fungsional. Yang dibutuhkan dalam sistem kebutuhan fungsional yaitu menjelaskan hal-hal apa saja yang dibutuhkan pada sistem agar sistem yang dibangun dapat berjalan sesuai dengan kebutuhan. Sedangkan untuk kebutuhan non fungsional pada sistem ini menjelaskan mengenai hal yang menjadi Batasan terhadap kebutuhan perancangan sistem. Berikut definisi kebutuhan sistem :

3.1.2.1 Kebutuhan Fungsional

Terdapat beberapa kebutuhan fungsional yang terdapat disistem, berikut merupakan kebutuhan fungsional :

1. Sistem dapat mengetahui keadaan darurat saat pengguna terjatuh ataupun saat pengguna berteriak dalam keadaan darurat.
2. Sistem dapat melakukan perhitungan logika *fuzzy* sesuai dengan *input* dari sensor *Gyroscope* dan sensor suara.
3. Sistem dapat menerapkan aturan yang terdapat pada logika *fuzzy*.
4. Sistem dapat menghasilkan *output* berupa notifikasi dan posisi terakhir *user* sistem yang akan dikirim ke *user* melalui aplikasi *mobile web*.

3.1.2.2 Kebutuhan Non-Fungsional

Didalam kebutuhan non-fungsional terdapat apa saja yang dibutuhkan sistem yaitu perangkat keras maupun perangkat lunak.

3.1.2.3 Kebutuhan Perangkat Keras

Terdapat beberapa kebutuhan dari perangkat keras yaitu sebagai berikut :

1. Mikrokontroler Arduino Mega 2560
Untuk mengolah dan memproses data pada sensor *Gyroscope*, sensor suara, modul ESP dan modul GPS.
2. Sensor *Gyroscope*
Untuk mendapatkan data dari kemiringan seseorang.
3. Sensor Suara
Untuk mendapatkan data level suara dari suara manusia.
4. Modul ESP
Untuk mengirim hasil data yang sudah diproses dan data dari *output* module GPS kemudian dikirim ke aplikasi *web*.
5. Module GPS
Untuk mendapatkan posisi pengguna saat keadaan darurat lalu data akan dikirim oleh modul ESP ke aplikasi *mobile web*.
6. Power 5V
Digunakan sebagai power dari kontroler.
7. Hp/PC
Digunakan sebagai sistem monitoring aplikasi *mobile web*.

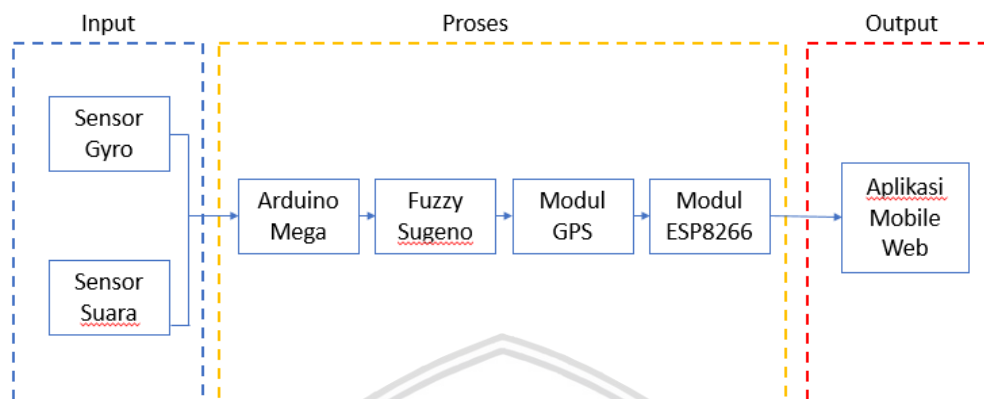
3.1.2.4 Kebutuhan Perangkat Lunak

Terdapat beberapa kebutuhan dari perangkat lunak yaitu sebagai berikut :

1. Sistem Operasi
Windows 10 sebagai OS untuk menjalankan Arduino IDE dan memprogram aplikasi *mobile web*.
2. Arduino IDE
Tempat untuk melakukan sebuah pemrograman pada sistem agar sistem berjalan sesuai dengan kebutuhan.
3. *Fuzzy*
Untuk penentu keputusan *output* data yang diambil dari data sensor.
4. Sublime Text 3
Tempat untuk melakukan pemrograman pada aplikasi *mobile web*.
5. MySQL
Tempat untuk sebuah server database open source yang digunakan sebagai aplikasi membuat *web*.
6. Server
Tempat untuk melakukan update, menyimpan data atau hapus data, dan dapat melihat seluruh data yang ada dalam database MySQL

3.1.3 Perancangan Sistem

Tahap ini bertujuan agar perancangan penelitian yang dilakukan menjadi terstruktur. Perancangan sistem penelitian ini tertera pada Gambar 3.4



Gambar 3.4 Diagram blok Perancangan Sistem

Terdapat 3 blok *input* yaitu sensor *Gyroscope*, sensor suara, dan tombol *button*. Kemudian dari *input* tersebut diproses di Arduino Mega, dari data yang didapat Arduino kemudian mengolah *fuzzy sugeno* serta menyertakan data yang didapat module GPS dan mengirimnya ke aplikasi *mobile web* menggunakan module ESP8266.

3.1.4 Implementasi

Pada tahap ini akan diimplementasikan seluruh gagasan serta ide baik desain maupun secara perhitungan sistem yang telah menjadi target tujuan sebelumnya. Terdapat beberapa tahap implementasi yaitu:

1. Implentasi *Hardware*

Dalam taham ini peneliti akan merancang desain *hardware* dipinggang bagian kiri karena sistem dari sensor *Gyroscope* memerlukan tempat yang statis dan stabil saat pengguna bergerak dengan normal, dan peneliti menghindari tempat desain yang akan mengakibatkan adanya gangguan yang kemungkinan tidak terjadi kondisi darurat terhadap *user*.

2. Implementasi *Software*

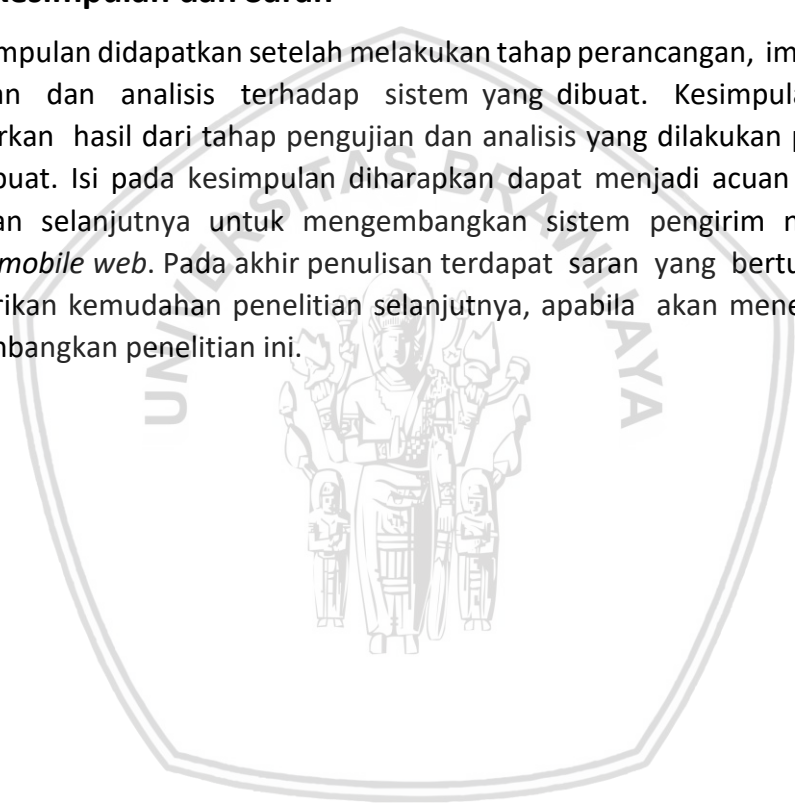
Didalam aplikasi *mobile web* peneliti hanya menggunakannya sebagai notifikasi untuk *user* dan dengan desain menu yang menarik.

3.1.5 Pengujian dan Analisis

Pengujian akan dilakukan kepada keseluruhan sistem yaitu dengan menampilkan notifikasi *output* sistem ke aplikasi *mobile web*. *Output* diberikan melalui perhitungan *fuzzy* berdasarkan data yang telah diterima pada sensor yang kemudian dikirimkan ke pusat kontroler (Arduino MEGA). Terdapat 4 pengujian yang akan dilakukan, yaitu pengujian akuisisi data sensor *Gyroscope*, akuisisi data sensor suara, akuisisi data *push button*, dan proses *fuzzy*. Data yang didapatkan dari hasil pengujian akan ditampilkan dalam bentuk notifikasi melalui aplikasi *mobile web*.

3.1.6 Kesimpulan dan Saran

Kesimpulan didapatkan setelah melakukan tahap perancangan, implementasi, pengujian dan analisis terhadap sistem yang dibuat. Kesimpulan disusun berdasarkan hasil dari tahap pengujian dan analisis yang dilakukan pada sistem yang dibuat. Isi pada kesimpulan diharapkan dapat menjadi acuan dasar pada penelitian selanjutnya untuk mengembangkan sistem pengirim notifikasi ke aplikasi *mobile web*. Pada akhir penulisan terdapat saran yang bertujuan untuk memberikan kemudahan penelitian selanjutnya, apabila akan meneruskan dan mengembangkan penelitian ini.



BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan menjelaskan persyaratan minimum yang harus terpenuhi agar perancangan dan implementasi sistem dapat berjalan dengan baik dan sesuai dengan apa yang sudah diharapkan sebelumnya.

4.1 Gambaran Umum Sistem

Bagian ini menjelaskan tentang analisis kebutuhan secara fungsional dan non fungsional yakni seperti *hardware* dan *software*. Kebutuhan secara *software* merupakan tahapan awal bagi peneliti untuk dapat memulai pemrograman menggunakan Arduino Mega yang berbasis Bahasa C terhadap *hardware* yang digunakan. Pada implementasi ini peneliti membuat system yang berukuran 9 x 10 cm dan tinggi 6 cm yang nantinya akan dipasang di pinggang sebelah kiri maupun kanan. Pada sistem ini terdapat 1 tombol *push button* yang berfungsi sebagai *trigger* dan jika pengguna sistem terjatuh maupun *user* berteriak dengan *input* level suara tinggi maka akan mengirim notifikasi ke aplikasi *mobile web* berupa pesan emergency dan lokasi keberadaan *user* saat mengalami keadaan darurat. Didalam aplikasi *mobile web*, keluarga terdekat akan mengetahui keberadaan user dengan melihat melalui google map yang sudah terhubung dengan aplikasi *mobile web*.

4.2 Analisis Kebutuhan

Untuk melakukan analisis kebutuhan sistem ini akan dibagi menjadi 2 seperti kebutuhan *user* dan kebutuhan sistem. Untuk membangun sistem ini memerlukan kebutuhan yang diperlukan agar dapat dipenuhi. Pada analisis kebutuhan *user* menjelaskan tentang kebutuhan fungsional dan juga kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Berikut adalah beberapa kebutuhan fungsional dari sistem yang akan dipenuhi, yaitu :

1. Sistem dapat mengetahui sudut kemiringan *user* saat terjatuh dan dapat mengetahui level *volume* suara *user* saat berteriak atau saat berbicara normal.
 - a. Penjelasan

Fungsi ini berguna untuk mendapatkan *input* nilai pada sensor *Gyroscope* dan sensor suara. Sensor *Gyroscope* berfungsi sebagai pendeteksi sudut kemiringan *user* saat terjatuh. Sedangkan sensor suara untuk mengetahui level *volume* suara *user* untuk mengetahui kondisi *user* memerlukan bantuan dengan nada suara tinggi atau sedang baik-baik saja dengan nada suara normal.

b. Respon/stimulus sistem

Pada saat sistem mendapatkan *input* nilai dan kondisi nilai sesuai pada pengambil keputusan logika *fuzzy* kemudian menandakan bahwa *user* dalam keadaan darurat, maka sistem akan mengirim notifikasi yang memberitahu bahwa *user* sedang mengalami kondisi yang darurat serta sistem akan mengirim posisi *user* ke aplikasi *mobile web* menggunakan module GPS yang sudah terhubung di sistem.

c. Kebutuhan Fungsional

Fungsi ini harus terpenuhi oleh sistem dikarenakan menjadi *input* pada sistem. Nilai *input* dari sistem ini akan dikomputasikan oleh Arduino Mega(Pusat Kontroler) untuk membuat fuzzifikasi dari sistem sebagai salah satu proses dalam *fuzzy* sugeno.

2. Sistem dapat melakukan perhitungan *fuzzy* sesuai dengan *input* dari sensor *Gyroscope* dan sensor suara.

a. Penjelasan

Fungsi ini berguna untuk sistem agar dapat melakukan perhitungan *fuzzy* mulai dari fuzzifikasi untuk mendapatkan fungsi keanggotaan dari masing-masing himpunan variabel, perhitungan inferensi untuk mendapatkan nilai α -predikat, dan defuzzifikasi untuk menentukan *output* dengan metode max. prioritas ini tinggi karena perhitungan *fuzzy* merupakan bagaimana *output* dari sistem.

b. Respon/Stimulus sistem

Saat sistem mendapatkan *input* pada kedua sensor, maka sistem akan melakukan perhitungan fuzzifikasi untuk mendapatkan nilai pada fungsi keanggotaan sesuai dengan rumus. Ketika nilai fungsi keanggotaan sudah didapatkan, maka sistem akan melakukan pengecekan pada setiap rule yang sesuai dengan nilai keanggotaan kemudian membuat inferensi untuk mendapatkan α -predikat dengan metode min untuk mendapatkan nilai keanggotaan baru hasil dari operasi 2 himpunan. Setelah diduplikatnya nilai keanggotaan baru, sistem akan melakukan defuzzifikasi dengan menggunakan metode max untuk menentukan hasil output dari sistem.

c. Kebutuhan Fungsional

Fungsi ini harus dipenuhi oleh sistem karena *output* sistem tergantung pada rule yang sudah dibuat. Rule yang sudah dibuat haruslah lengkap sesuai dengan himpunan *fuzzy* maupun nilai data yang sudah diambil dari data latih, karena jika salah satu rule tidak lengkap maka *output* yang keluar akan tidak sesuai kebutuhan sistem.

3. Sistem dapat menerapkan aturan *fuzzy*.

a. Penjelasan

Fungsi ini dapat diharapkan agar sistem bisa menerapkan aturan-aturan yang terdapat pada logika *fuzzy*, sehingga sistem dapat mengeluarkan *output* yang sesuai dengan kebutuhan berupa notifikasi yang berada di aplikasi *mobile web*.

b. Respon/stimulus sistem

Pada saat sistem telah melakukan perhitungan untuk mendapatkan nilai keanggotaan dalam proses fuzzifikasi, maka sistem selanjutnya akan melakukan pengecekan pada setiap rule/aturan yang sesuai dengan nilai keanggotaan. Setelah rule ditemukan, maka sistem akan melakukan inferensi dengan menentukan nilai α -predikat menggunakan metode min untuk mendapatkan nilai keanggotaan baru dari hasil operasi 2 himpunan.

c. Kebutuhan Fungsional

Fungsi ini harus bisa dipenuhi oleh sistem karena *output* sistem bergantung pada rule yang dibuat. Rule yang telah dibuat harus lengkap sesuai dengan himpunan *fuzzy* yang ada, karena jika salah satu rule tidak lengkap maka *output* yang dikeluarkan oleh sistem tidak sesuai.

4. Sistem dapat menghasilkan *output* berupa notifikasi yang akan dikirim ke aplikasi android sesuai dengan perhitungan *fuzzy*.

a. Penjelasan

Fungsi ini dapat diharapkan agar sistem bisa memberikan notifikasi *output* pada aplikasi *mobile web* agar *user* dapat lebih mudah mengamati pengguna sistem.

b. Respon/stimulus sistem

Pada saat seluruh proses *fuzzy* telah dilakukan dan sistem dapat menentukan *output*nya, maka sistem dapat mengirim notifikasi pesan ke aplikasi *mobile web*.

c. Kebutuhan Fungsional

Fungsi ini berguna untuk memberikan *output* dari sistem sebagai penerima notifikasi pesan yang berada di aplikasi *mobile web*.

4.2.2 Kebutuhan Non-Fungsional

Didalam kebutuhan non-fungsional terbagi menjadi 2 kebutuhan, yaitu kebutuhan perangkat keras dan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Terdapat berapa kebutuhan perangkat keras untuk membuat sistem ini, diantaranya adalah :

1. Arduino Mega 2560

Arduino Mega merupakan sebuah mikrokontroler yang berperan sebagai pusat kontroler sebagai inti dari pemrosesan dari prototipe sistem yang dibuat. Arduino Mega digunakan penulis oleh penulis sebagai pusat kontroler dikarenakan Arduino Mega cukup untuk melakukan proses *fuzzy*.

2. Sensor *Gyroscope*

Sensor *Gyroscope* merupakan sensor yang berperan sebagai pengukur sudut kemiringan pada kondisi manusia agar dapat mengetahui kondisi manusi itu sedang baik-baik saja dengan berdiri normal atau sedang keadaan darurat seperti terjatuh. Kemiringan sudut pada sensor *Gyroscope* yaitu antara -180,00 sampai 180,00.

3. Sensor Suara

Sensor suara akan berperan sebagai pengukur level *volume* suara manusia, seperti saat suara tinggi atau manusia sedang teriak meminta pertolongan maka sensor ini akan terolah dengan sensor lain sebagai *input* dan di proses melalui logika *fuzzy* sensor ini akan mengirimkan sebuah notifikasi ke pihak keluarga melalui aplikasi *mobile web*.

4. Modul ESP

Modul ini berfungsi sebagai perantara antara *input* dan *output* sebagai jaringan yang berguna untuk mengirim sebuah sinyal notifikasi ke aplikasi *mobile web*.

5. Modul GPS

Modul ini berperan sebagai pemberi lokasi terakhir *user* saat keadaan darurat. Lokasi didapatkan dari kedua sensor yang sudah diolah oleh *fuzzy* kemudian lokasi akan dikirimkan ke keluarga melalui aplikasi *mobile web*.

6. Perangkat Laptop/Handpone

Perangkat laptop/Handphone dalam prototipe ini digunakan sebagai media untuk membuat program yang digunakan Arduino Mega. Selain itu laptop/handphone juga digunakan sebagai aplikasi *mobile* yang menerima sebuah notifikasi saat *user* dalam keadaan darurat.

4.2.2.2 Kebutuhan Perangkat Lunak

Untuk Kebutuhan perangkat lunak yang dibutuhkan dalam prototipe sistem ini adalah :

1. Arduino IDE

Arduino IDE digunakan sebagai tempat pemrograman untuk prototipe sistem menggunakan Bahasa C++.

2. Sublime Text

Sublime text digunakan untuk tempat pemrogram untuk prototipe sistem di aplikasi *mobile web* ini.

3. XAMPP

Untuk membantu membuat sebuah webserver.

4. MySQL

Tempat untuk sebuah server *database open source* yang digunakan sebagai aplikasi membuat *web*.

5. Server

Tempat untuk melakukan update, menyimpan data atau hapus data, dan dapat melihat seluruh data yang ada dalam database MySQL

4.3 Batasan Desain Sistem

Pembuatan prototipe sistem notifikasi saat keadaan darurat dengan monitoring aplikasi *mobile web* terdapat adanya Batasan yang akan diterapkan, sehingga saat proses pembahasan sampai implementasi dari prototipe sistem ini tidak mencakup jangkauan sangat luas. Batasan-batasan desain sistem prototipe tersebut sebagai berikut :

1. Sistem menandakan on/off saat terdapat kabel yang tertancap maupun tercabut dari *input* daya.
2. Pemasangan implementasi sistem pada manusia terletak di pinggang sebelah kiri, karena dari sensor *Gyroscope* membutuhkan titik koordinat yang stabil agar saat sistem digunakan sudut nilai koordinat sesuai dengan kebutuhan.
3. Desain Sistem pada *input* sensor suara terdapat kabel tambahan yang dimana desain akan terpisah dari desain pusat karena sensor suara membutuhkan suara yang lebih dekat dengan *input* berupa *volume* suara.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

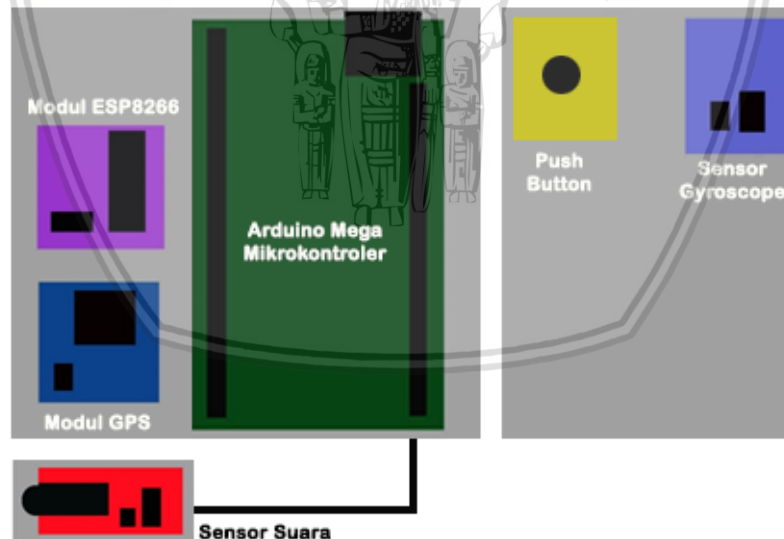
Pada bab ini akan menjelaskan proses perancangan dan implementasi sistem. Terdapat dua perancangan yaitu perancangan perangkat lunak (*Software*) dan perancangan perangkat keras (*Hardware*). Sedangkan pada implementasi terdapat implementasi perangkat lunak (*Software*) dan implementasi perangkat keras (*Hardware*).

5.1 Perancangan Sistem

Pada perangkat yang dirancang, sistem dapat mengirim notifikasi berdasarkan kondisi pengguna dari kemiringan dan level *volume* suara ketika keadaan darurat. Untuk *input* berupa kemiringan dan *volume* suara pengguna yang sudah diolah menggunakan mikrokontroler. Sedangkan *output* pada sistem menggunakan metode *fuzzy sugeno* untuk menentukan apakah pengguna sistem terjatuh atau berteriak saat dalam keadaan darurat.

5.1.1 Perancangan Alat Mekanik, Peletakan dan Ukuran sistem

Dalam perancangan alat mekanik, peletakan dan ukuran menggunakan sensor *Gyroscope*, sensor suara, modul GPS, modul ESP8266, dan limit switch yang diletakan pada kotak berbahan plastik yang akan dilapisi kain atau spon.

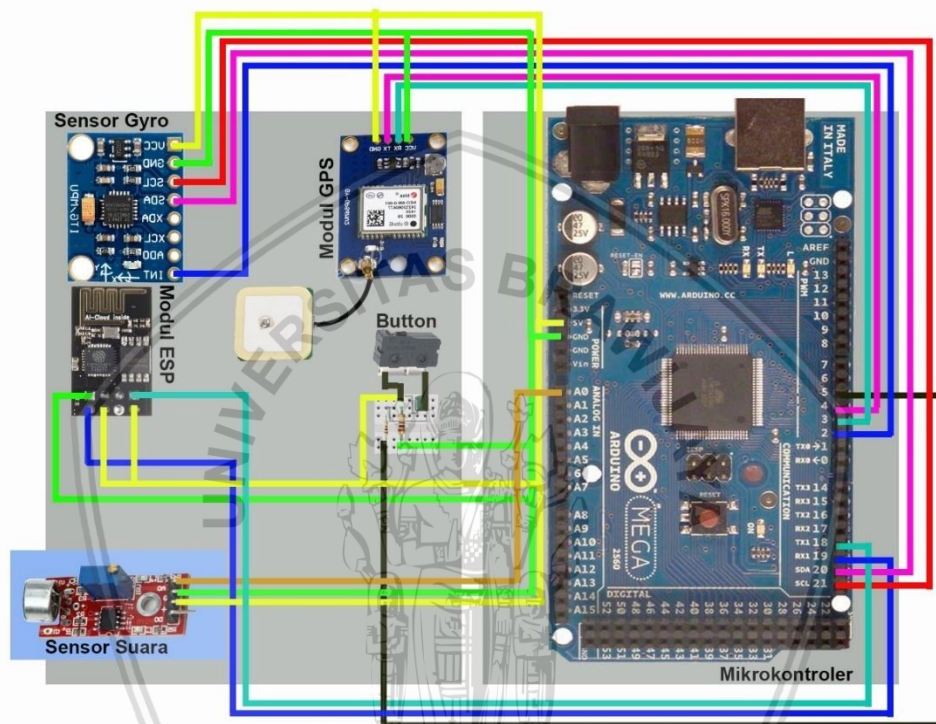


Gambar 5.1 Skema Perancangan Sistem

Pada Gambar 5.1 dijelaskan bahwa bagian posisi samping box sistem. Terdapat juga box kusus untuk sensor suara, box untuk sensor suara tidak tergabung menjadi satu dengan box pusat, karena sensor suara membutuhkan *input* suara yang harus dekat dengan *input* sensor suara.

5.1.2 Perancangan Perangkat Keras

Perangkat keras (*Hardware*) digunakan untuk membangun sebuah sistem notifikasi yang akan mendukung mikronkontroler untuk menerapkan logika *fuzzy* sebagai pengambil keputusan *output*. Mikrokontroler yang digunakan adalah Arduino Mega. *Input* yang digunakan untuk mengukur sudut kemiringan manusia adalah sensor *Gyroscope* dan sensor suara untuk mengukur level *volume* suara pada manusia.



Gambar 5.2 Skema Perancangan Perangkat Keras

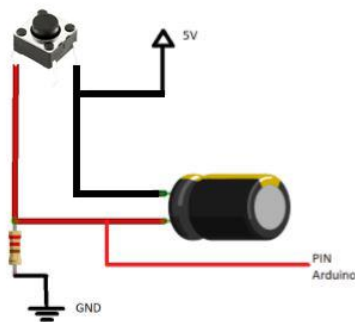
Untuk memperjelas dari Gambar 5.2 rangkaian perangkat keras dari pin-pin yang tersambung dari Arduino Mega ke sensor maupun modul seperti Tabel 5.1.

Tabel 5.1 Koneksi Pin ke Mikrokontroler di Perancangan Perangkat Keras

Arduino Mega	Modul GPS	Modul ESP	Sensor Suara	Sensor Gyroscope	Push Button
2				INT	
3	RX				
4	TX				
7					PIN
18 RX		TX			
19TX		RX			
20 SDA				SDA	
21 SCL				SCL	
A0			A0		
GND	GND	GND	GND	GND	GND
5V	VCC	VCC & CH-PD	+	VCC	VCC

Pada Gambar 5.2 diatas, *input* data berupa sensor suara dan sensor *Gyroscope*. Pada data sensor suara dan sensor *Gyroscope* yang telah diambil kemudian di proses pada Arduino Mega sehingga sistem akan bisa mengolah melalui *fuzzy* yang akan menentukan *output* data yaitu normal atau *emergency*, kemudian *output* akan dikirim menggunakan modul ESP8266 dengan method post yang akan ditampilkan oleh aplikasi *mobile web* serta lokasi titik koordinat user yang didapatkan melalui modul GPS.

Pada *push button* terdapat adanya rangkaian *debounce* yang berguna untuk menghilangkan efek *bouncing* yang terjadi pada tombol *push button*, karena efek *bouncing* menyebabkan nilai dari variable *push button* tidak akurat dan tidak berjalan dengan semestinya. Saat *push button* ditekan value terkadang tidak menentu antara 0 atau 1, maka dari itu diperlukannya rangkaian *debounce*. Berikut Gambar 5.3 ini adalah rangkaian *debounce* yang terhubung pada *push button*.

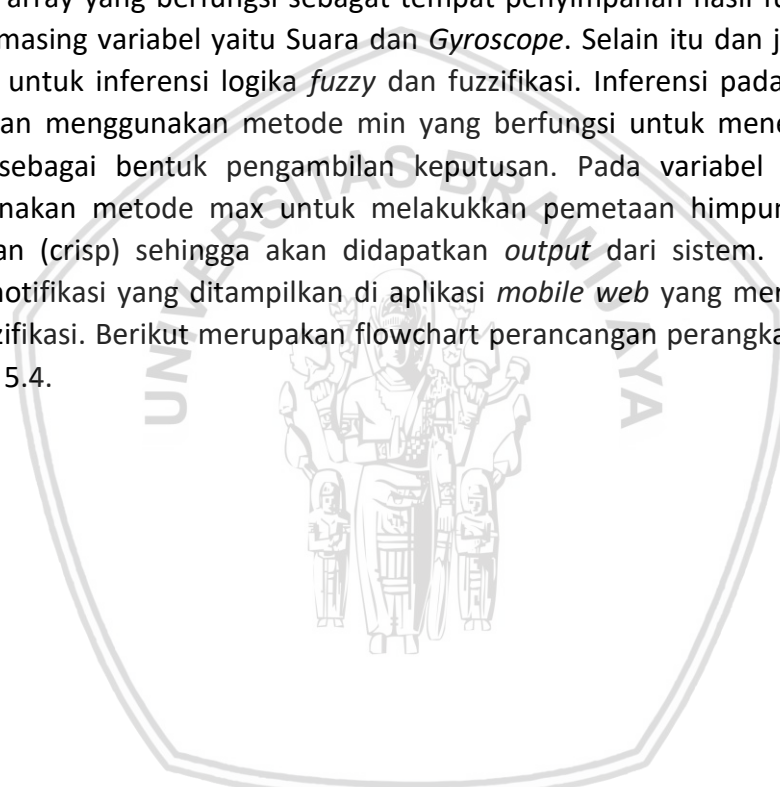


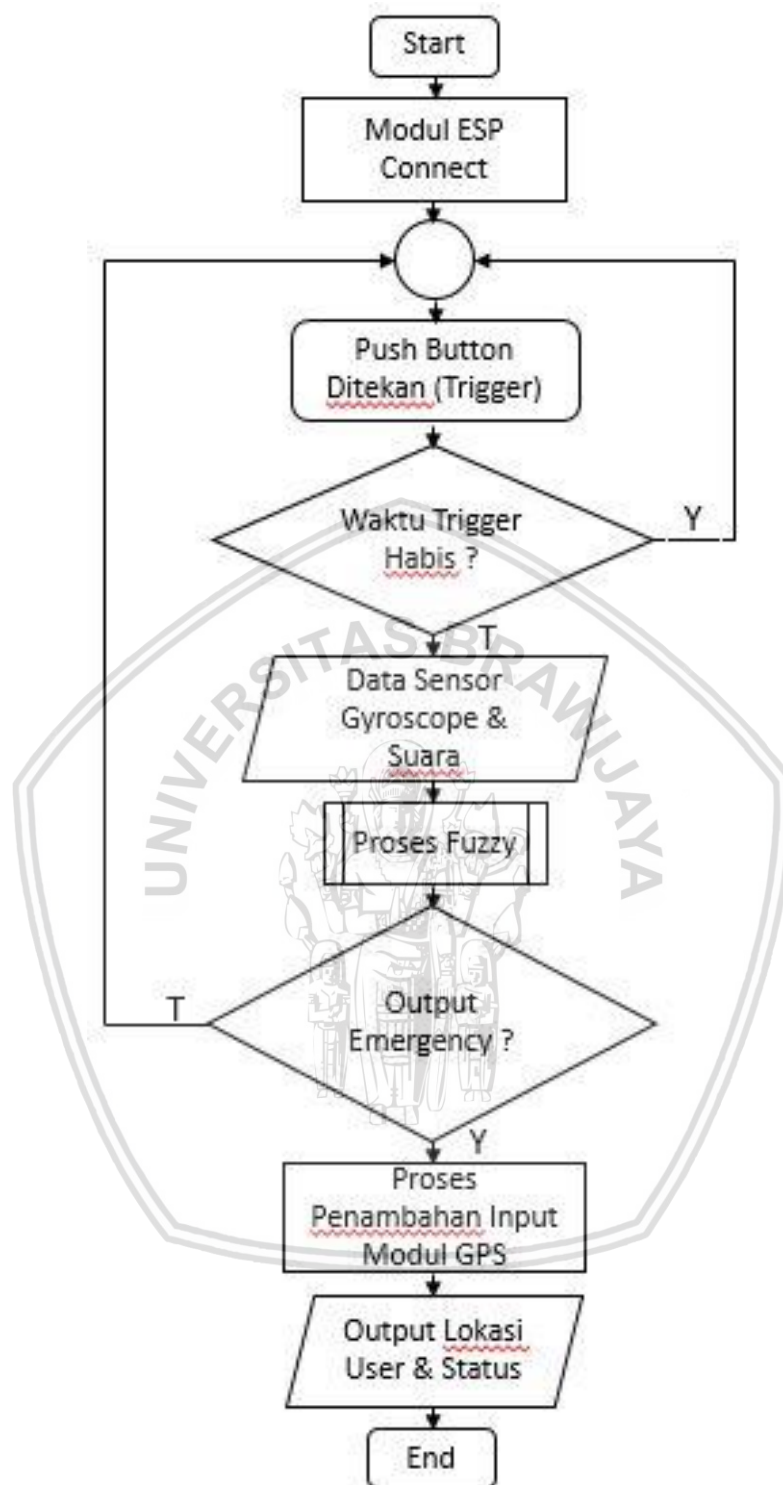
Gambar 5.3 Rangkaian *Debounce*

5.1.3 Perancangan Perangkat Lunak

Pada sistem ini untuk perangkat lunak berupa code program yang digunakan untuk mikrokontroler yaitu Arduino IDE. Code Program yang dibuat dengan menerapkan metode logika *fuzzy* sebagai pengambil keputusan untuk *output* berupa notifikasi yang akan ditampilkan melalui aplikasi *mobile web* yang berdasarkan dari *input* sensor *Gyroscope* untuk mengukur kemiringan pengguna dan sensor suara untuk mengukur level *volume* suara pengguna. Data dari 2 sensor tersebut akan disimpan pada suatu *variable* yang akan diproses untuk menentukan himpunan logika *fuzzy* dari masing-masing variabel.

Implementasi *fuzzy* sugeno ke dalam sistem notifikasi ini akan dibuat suatu variabel array yang berfungsi sebagai tempat penyimpanan hasil fuzzifikasi dari masing-masing variabel yaitu Suara dan *Gyroscope*. Selain itu dan juga terdapat variabel untuk inferensi logika *fuzzy* dan fuzzifikasi. Inferensi pada logika *fuzzy* dilakukan menggunakan metode min yang berfungsi untuk menentukan nilai *output* sebagai bentuk pengambilan keputusan. Pada variabel defuzzifikasi menggunakan metode max untuk melakukan pemetaan himpunan *fuzzy* ke himpunan (crisp) sehingga akan didapatkan *output* dari sistem. *Output* pada sistem notifikasi yang ditampilkan di aplikasi *mobile web* yang merupakan hasil dari fuzzifikasi. Berikut merupakan flowchart perancangan perangkat lunak pada Gambar 5.4.

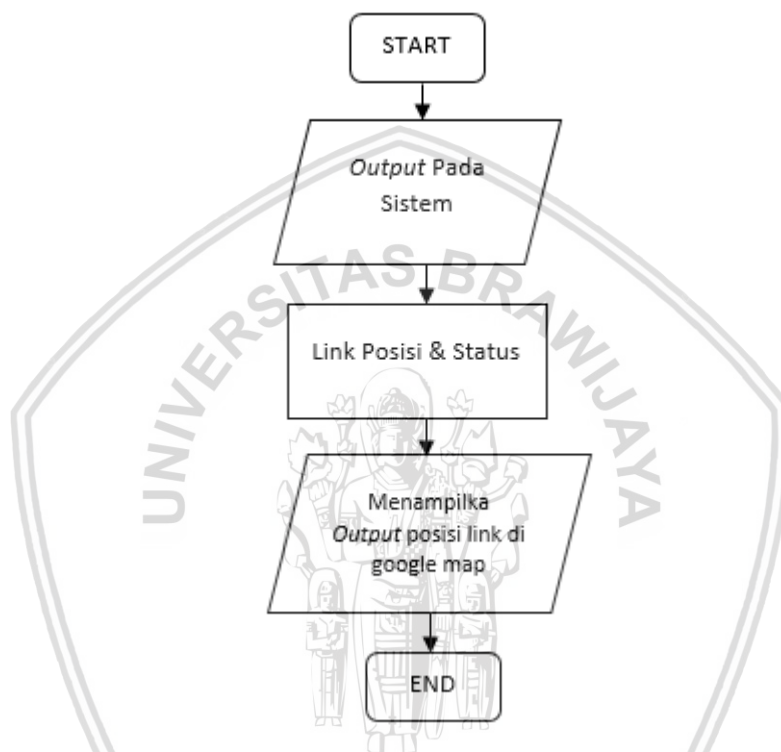




Gambar 5.4 Flowchart Perancangan Perangkat Lunak

5.1.3.1 Perancangan Aplikasi *Mobile Web*

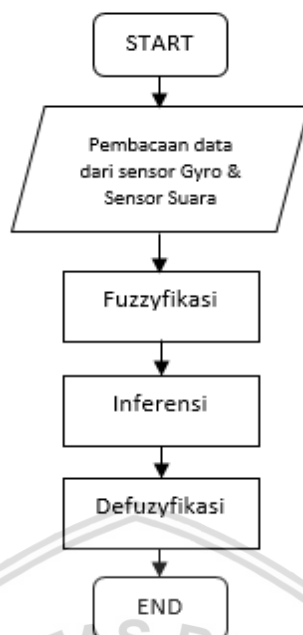
Pada perancangan aplikasi *mobile web*, keutamaan sistem ini memiliki fungsi untuk menampilkan notifikasi berupa pesan yang berisi keterangan keadaan darurat *user* sistem dan posisi *user* terakhir kali sistem mendapatkan lokasi keberadaan melalui modul GPS. Tampilan ini terdapat di dalam menu aplikasi *inbox* berdasarkan hasil *output* yang dikeluarkan oleh sistem yang akan dijelaskan pada Gambar 5.5 dibawah ini yang merupakan fungsi utama kegunaan aplikasi *mobile web*.



Gambar 5.5 Flowchart Perancangan Kegunaan Utama Aplikasi *Mobile Web*

5.1.3.2 Perancangan Fuzzy

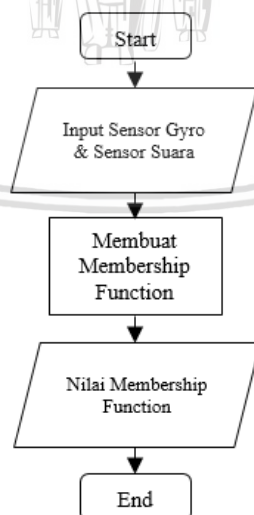
Didalam sistem perancangan menggunakan metode *fuzzy* membutuhkan proses hingga didapatkannya keputusan *output* dari sistem yang sesuai dengan perhitungan *fuzzy*. Proses di metode *fuzzy* yaitu berupa fuzzifikasi, pembuatan rule data, inferensi, serta defuzzifikasi. Gambar 5.6 menunjukkan tentang sub-proses control *fuzzy* yang memiliki fungsi sebagai pengambil keputusan untuk mendapatkan *output* sesuai dengan kebutuhan sistem. Berikut ini adalah flowchart proses perancangan *fuzzy* :



Gambar 5.6 Flowchart Proses Perancangan *Fuzzy*

1. Fuzzifikasi

Fuzzifikasi yaitu suatu pengubahan proses nilai tegas (Crisp) didalam fungsi keanggotaan *fuzzy*. Gambar 5.7 merupakan sebuah sub-proses fuzzifikasi yang akan memproses *input* dari sensor *Gyroscope* untuk mendapatkan kemiringan kondisi manusia, dan sensor suara sebagai *input* berupa level *volume* pada suara manusia. Fungsi sub-proses fuzzifikasi yaitu mengubah crisp/nilai tegas kedalam fungsi keanggotaan *fuzzy*. Berikut Gambar 5.7 adalah flowchart proses fuzzifikasi.



Gambar 5.7 Flowchart Proses Fuzzifikasi

Implementasi yang terdapat di Metode *Fuzzy Sugeno* pada sistem notifikasi saat terjadi keadaan darurat kepada pengguna memiliki 2 *input* yaitu suara dan *Gyroscope*, pada variabel *Gyroscope* dibagi menjadi 3 himpunan yaitu *Gyroscope* Y, *Gyroscope* P, dan *Gyroscope* R. Sedangkan untuk variabel suara dibagi menjadi 4 himpunan yaitu rendah, normal, sedang, tinggi, dan sangat tinggi.

a. Variabel Suara

Pada variabel Suara, penulis menggunakan 4 himpunan *fuzzy* yaitu rendah, sedang, tinggi, dan sangat tinggi. Berikut merupakan himpunan di variabel suara.

- Rendah [0 19]

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[\text{Rendah}] = \begin{cases} 1, x \leq 16 \\ \frac{19-x}{19-16}, 16 < x \leq 19 \\ 0, x \geq 19 \end{cases}$$

- Sengah [16 35]

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[\text{Sengah}] = \begin{cases} 1, 19 < x \leq 27 \\ \frac{35-x}{35-27}, 27 < x \leq 35 \\ \frac{x-16}{19-16}, 16 < x \leq 19 \\ 0, x \geq 35 \end{cases}$$

- Tinggi [27 352]

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[\text{Tinggi}] = \begin{cases} 1, 35 < x \leq 200 \\ \frac{352-x}{352-200}, 200 < x \leq 352 \\ \frac{x-27}{35-27}, 27 < x \leq 35 \\ 0, x \geq 352 \end{cases}$$

- Sangat Tinggi [> 200]

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[\text{Sangat Tinggi}] = \begin{cases} 1, x \geq 352 \\ \frac{x-200}{325-200}, 200 < x \leq 352 \\ 0, x \leq 200 \end{cases}$$

Fungsi dalam keanggotaan himpunan *fuzzy* rendah, sedang, tinggi, dan sangat tinggi dari variabel suara direpresentasikan seperti Gambar 5.8.



Gambar 5.8 Membership Function Suara

b. Variabel Gyroscope

Dalam percobaan pengukuran sensor *Gyroscope* dengan melihat data nilai pada serial yang terdapat di monitor Arduino IDE. Percobaan dilakukan dengan 2 kondisi, terjatuh dan normal dengan cara men-on/aktifkan Arduino secara berulang-ulang sebanyak 100 kali dan menjatuhkan sensor dengan posisi terjatuh kedepan, belakang, kanan, dan kiri yang masing-masing sebanyak 20 kali percobaan secara berulang-ulang.

- *Gyroscope Y*

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[YBeridiri] = \begin{cases} 1, & 0 < x \leq 16 \\ \frac{x - (-62)}{0 - (-62)}, & -62 < x \leq 0 \\ \frac{65 - x}{65 - 16}, & 16 < x \leq 65 \\ 0, & x \geq 65 \end{cases}$$

$$\mu[YTerjatuh] = \begin{cases} 1, & -118 < x \leq -62 \\ \frac{0 - x}{0 - (-62)}, & -62 < x \leq 0 \\ 0, & x \geq 0 \\ 1, & x \geq 65 \\ \frac{x - 16}{65 - 16}, & 16 < x \leq 65 \\ 0, & x < 16 \end{cases}$$

- *Gyroscope P*

Fungsi keanggotaannya dirumuskan sebagai berikut :

$$\mu[\text{PBerdiri}] = \begin{cases} 1, & 0 < x \leq 2 \\ \frac{x - (-62)}{0 - (-62)}, & -62 < x \leq 0 \\ \frac{46 - x}{46 - 2}, & 2 < x \leq 46 \\ 0, & x \geq 46 \end{cases}$$

$$\mu[\text{PTerjatuh}] = \begin{cases} 1, & -86 < x \leq -62 \\ \frac{0 - x}{0 - (-62)}, & -62 < x \leq 0 \\ 1, & x \geq 46 \\ \frac{x - 2}{46 - 2}, & 2 < x \leq 46 \\ 0, & 2 < x \end{cases}$$

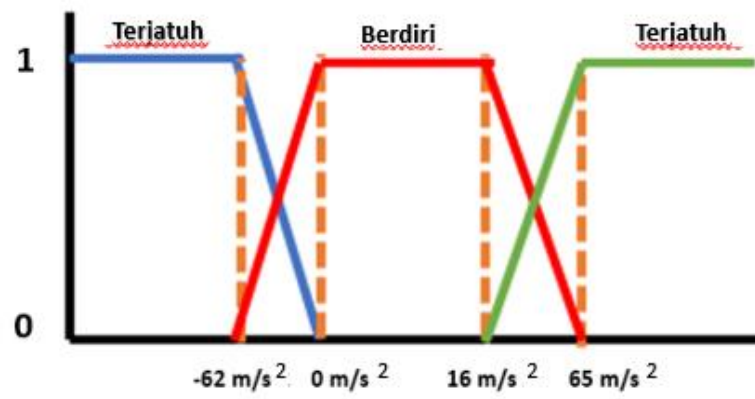
- *Gyroscope R*

Fungsi keanggotaannya dirumuskan sebagai berikut :

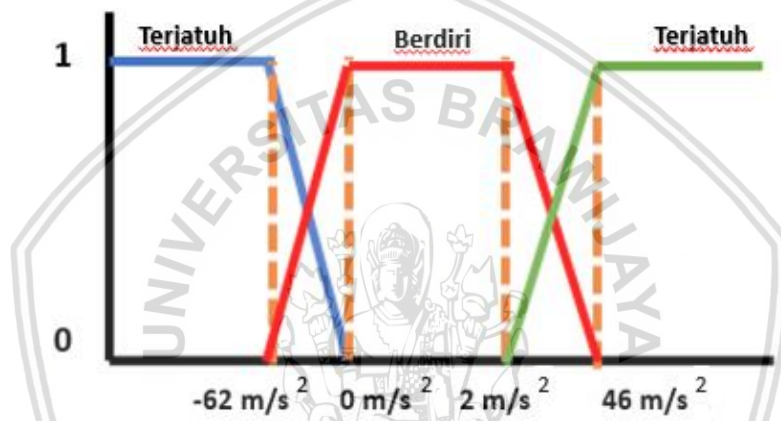
$$\mu[\text{RBerdiri}] = \begin{cases} 1, & x \geq 29 \\ \frac{x - 20}{29 - 20}, & 20 < x \leq 29 \\ 0, & x \leq 20 \end{cases}$$

$$\mu[\text{RTerjatuh}] = \begin{cases} 1, & -20 < x \leq 20 \\ \frac{29 - x}{29 - 20}, & 20 < x \leq 29 \\ 0, & x \geq 29 \end{cases}$$

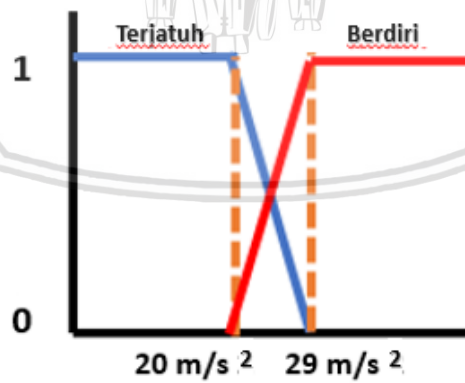
Fungsi dalam keanggotaan himpunan *fuzzy* terjatuh dan berdiri dari variabel *Gyroscope* direpresentasikan seperti Gambar 5.9 pada *Gyroscope Y*, gambar 5.10 pada *Gyroscope P*, dan gambar 5.11 pada *Gyroscope R*.



Gambar 5.9 Membership Function Gyroscope Y



Gambar 5.10 Membership Function Gyroscope P



Gambar 5.11 Membership Function Gyroscope R

2. Pembuatan Rule

Rule merupakan bentuk implikasi if-then (jika-maka). Implikasi yaitu membentuk aturan yang berupa implikasi *fuzzy* untuk menyatakan relasi antara *input* dan *output*. Di metode sugeno, implikasi yang digunakan yaitu min. Pada pembuatan aturan/rule menggunakan IF dan AND yang akan menghasilkan perintah "THEN". Adanya aturan *fuzzy* yaitu digunakan sebagai penentu *output* yang berupa notifikasi *emergency* atau tidak adanya notifikasi melalui aplikasi *mobile web*. Penjelasan aturan rule terdapat di tabel 5.2.

Tabel 5.2 Rule Logika Fuzzy

Rule Ke-	Suara	Gyro-Y	Gyro-P	Gyro-R	Output
0	Rendah	Berdiri	Berdiri	Berdiri	Normal
1	Rendah	Berdiri	Berdiri	Terjatuh	Normal
2	Rendah	Berdiri	Terjatuh	Berdiri	Normal
3	Rendah	Berdiri	Terjatuh	Terjatuh	Normal
4	Rendah	Terjatuh	Berdiri	Berdiri	Normal
5	Rendah	Terjatuh	Berdiri	Terjatuh	Normal
6	Rendah	Terjatuh	Terjatuh	Berdiri	Normal
7	Rendah	Terjatuh	Terjatuh	Terjatuh	Emergency
8	Sedang	Berdiri	Berdiri	Berdiri	Normal
9	Sedang	Berdiri	Berdiri	Terjatuh	Normal
10	Sedang	Berdiri	Terjatuh	Berdiri	Normal
11	Sedang	Berdiri	Terjatuh	Terjatuh	Normal
12	Sedang	Terjatuh	Berdiri	Berdiri	Normal
13	Sedang	Terjatuh	Berdiri	Terjatuh	Normal
14	Sedang	Terjatuh	Terjatuh	Berdiri	Normal
15	Sedang	Terjatuh	Terjatuh	Terjatuh	Emergency
16	Tinggi	Berdiri	Berdiri	Berdiri	Emergency
17	Tinggi	Berdiri	Berdiri	Terjatuh	Emergency
18	Tinggi	Berdiri	Terjatuh	Berdiri	Emergency
19	Tinggi	Berdiri	Terjatuh	Terjatuh	Emergency
20	Tinggi	Terjatuh	Berdiri	Berdiri	Emergency
21	Tinggi	Terjatuh	Berdiri	Terjatuh	Emergency
22	Tinggi	Terjatuh	Terjatuh	Berdiri	Emergency
23	Tinggi	Terjatuh	Terjatuh	Terjatuh	Emergency
24	Sangat Tinggi	Berdiri	Berdiri	Berdiri	Emergency
25	Sangat Tinggi	Berdiri	Berdiri	Terjatuh	Emergency
26	Sangat Tinggi	Berdiri	Terjatuh	Berdiri	Emergency

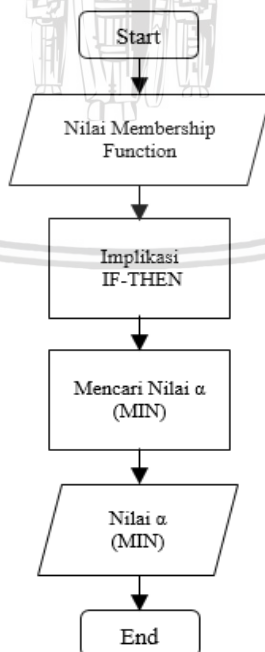
Rule Ke-	Suara	Gyro-Y	Gyro-P	Gyro-R	Output
27	Sangat Tinggi	Berdiri	Terjatuh	Terjatuh	Emergency
28	Sangat Tinggi	Terjatuh	Berdiri	Berdiri	Emergency
29	Sangat Tinggi	Terjatuh	Berdiri	Terjatuh	Emergency
30	Sangat Tinggi	Terjatuh	Terjatuh	Berdiri	Emergency
31	Sangat Tinggi	Terjatuh	Terjatuh	Terjatuh	Emergency

3. Inferensi

Inferensi merupakan proses dari implikasi untuk menalarkan nilai *input* yang akan menentukan nilai *output*/keluaran sebagai mode untuk pengambil keputusan. Berdasarkan dari pembuatan rule pada tabel 5.2 maka akan ditentukan sebuah nilai α pada setiap rule. α yaitu nilai keanggotaan yang baru dari hasil operasi 2 himpunan atau lebih. Rule yang akan digunakan adalah MIN pada fungsi implikasi. Rumus untuk mendapatkan nilai inferensi yaitu :

$$\mu A \cap B[x] = \min(\mu A[x], \mu B[x])$$

Berikut Gambar 5.12 adalah flowchart dari proses inferensi yang digunakan untuk mendapatkan nilai α sebagai nilai keanggotaan baru.



Gambar 5.12 Flowchart Proses Inferensi

Pada Gambar 5.12 menjelaskan tentang inferensi yang membutuhkan *input* yaitu berupa *membership function* yang didapatkan pada proses sub-proses di fuzzifikasi. Nilai yang terdapat di *membership function* akan dicocokkan dengan rule. jika cocok, maka akan diterapkan metode MIN untuk mencari nilai yang terkecil dari setiap rule. Sehingga akan didapatkan nilai α yang merupakan nilai keanggotaan baru.

Berikut ini merupakan contoh penggunaan dari inferensi. Sebuah perhitungan fuzzifikasi didapatkan nilai keanggotaan suara rendah 0, Gyro Y jatuh 1, Gyro P 0.9, dan Gyro R adalah 1. Maka hasil dari nilai keanggotaan baru adalah sebagai berikut :

- Rule[5] : IF SUARA RENDAH && GYRO Y TERJATUH && GYRO P BERDIRI && GYRO R TERJATUH

$$\alpha\text{-predikat1} = \min (\mu_{\text{SuaraRENDAH}} \cap \mu_{\text{YJATUH}} \cap \mu_{\text{PBERDIRI}} \cap \mu_{\text{RJATUH}})$$

$$= \min (0 \cap 1 \cap 0.9 \cap 1)$$

$$= 0$$
- Rule[8] : IF SUARA SEDANG && GYRO Y BERDIRI && GYRO P BERDIRI && GYRO R BERDIRI

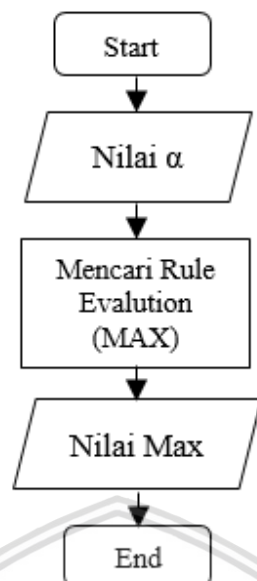
$$\alpha\text{-predikat1} = \min (\mu_{\text{SuaraSEDANG}} \cap \mu_{\text{YBERDIRI}} \cap \mu_{\text{PBERDIRI}} \cap \mu_{\text{RBERDIRI}})$$

$$= \min (0.7 \cap 0.7 \cap 0.9 \cap 1)$$

$$= 0.7$$

4. Defuzzifikasi

Defuzzifikasi merupakan sebuah proses pemetaan dari himpunan logika *fuzzy* ke himpunan *crisp*(tegas). Setelah didapatkan nilai α , selanjutnya yaitu berdasarkan metode MIN-MAX di setiap variabel maka akan dievaluasi di setiap rulanya untuk mencari nilai MAX (terbesar). Berikut Gambar 5.13 ini adalah flowchart yang terjadi di defuzzifikasi :



Gambar 5.13 Flowchart Proses Defuzzifikasi

Pada Gambar 5.13 menjaskan bahwa proses defuzzifikasi membutuhkan *input* yang berasal dari nilai α yaitu nilai keanggotaan baru. Nilai α diperoleh dari proses inferensi. Setelah α diperoleh, maka selanjutnya yaitu rule evaluation. Pada proses rule evaluation menggunakan metode MAX untuk mendapatkan nilai terbesar yang dijadikan *output* pada sistem. Terdapat adanya 1 *output* untuk defuzzifikasi. Berikut ini adalah cara untuk pengambilan nilai MAX untuk mendapatkan *output* dari sebuah sistem yang mendapatkan nilai Suara 44, Gyro Y 11.3, Gyro P -84.89, dan Gyro R yaitu -5.39 :

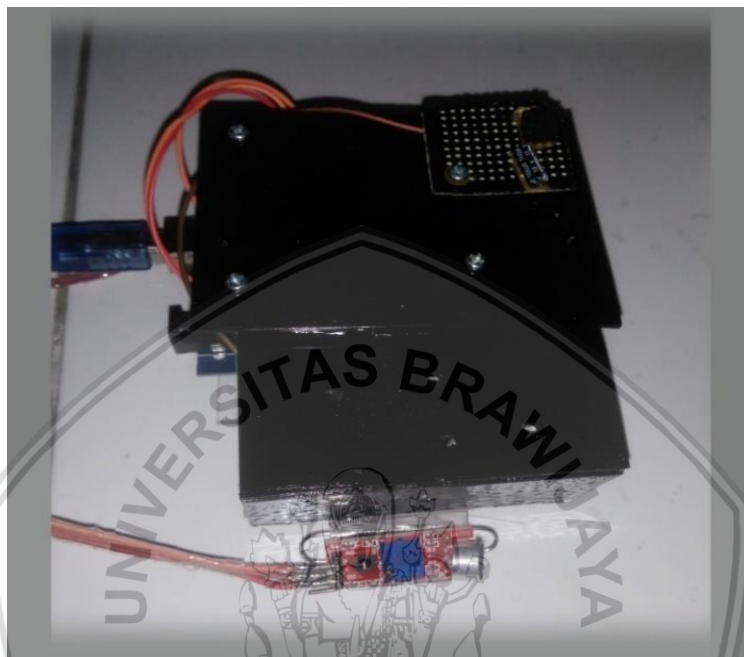
- $$\begin{aligned} \text{Output} &= \max(\text{rule0, rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8,} \\ &\quad \text{rule9, rule10, rule11, rule12, rule13, rule14, rule15, rule16,} \\ &\quad \text{rule 17, rule19, rule20, rule21, rule22, rule23, rule24, rule25,} \\ &\quad \text{rule26, rule27, rule28, rule29, rule31}) \\ &= \max(0, 1, 0, \\ &\quad 0, 0, 0, 0, 0, 0) \\ &= \max(1) \\ &= \text{rule 24 output Emergency} \end{aligned}$$

5.2 Implementasi Sistem

Pada implementasi sistem terdapat beberapa sub-bab yang akan menjelaskan tentang sistem pada perangkat keras dan perangkat lunak.

5.2.1 Implementasi Alat Mekanik, Peletakan dan Ukuran Sistem

Didalam tahap implementasi seperti yang telah disebutkan di sub bab 5.1.1. Alat yang digunakan sebagai tempat sistem berbahan akrilik yang berukuran panjang 10cm dan lebar 9cm sebagai alas maupun penutup alat. Untuk peletakan komponen-komponen akan diletakkan sesuai ukuran komponen agar ukuran alat bisa seminimalis mungkin.

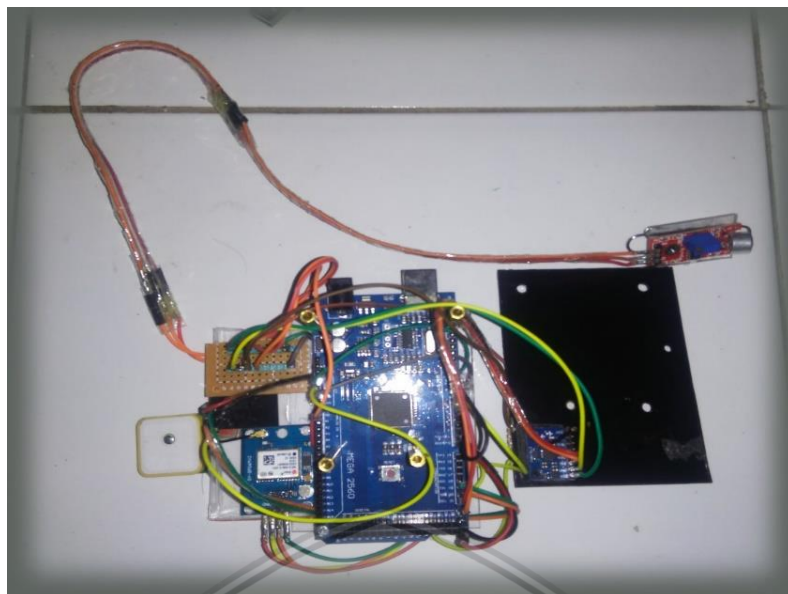


Gambar 5.14 Implementasi *Prototype* Sistem Tampak Samping

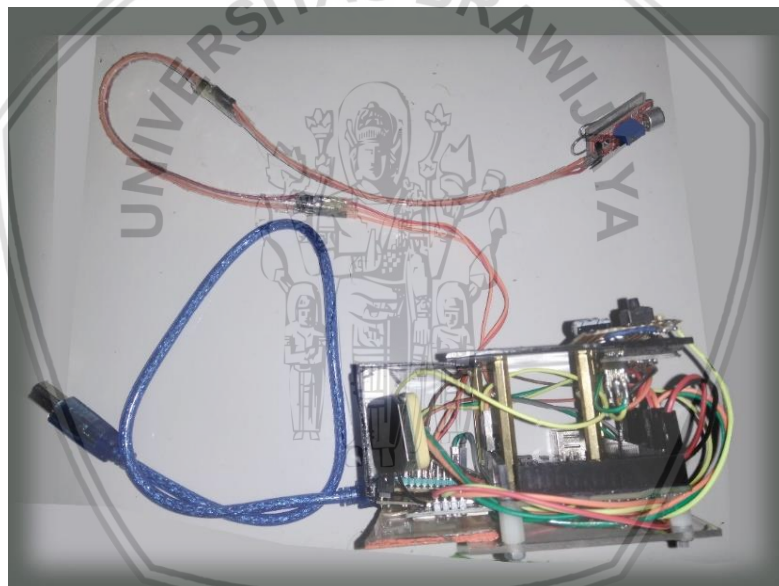
Gambar 5.14 merupakan implementasi prototipe sistem tampak samping. Terdapat sensor suara yang terlepas dari desain sistem utama, karena sensor suara membutuhkan masukan *input* yang lebih dekat dengan suara. Terdapat juga *push button* yang terletak disamping bagian atas desain sistem, ini bertujuan agar *push button* lebih dekat dengan tangan *user* yang berguna sebagai *trigger*.

5.2.2 Implementasi Perangkat Keras

Saat melakukan implementasi perangkat keras harus sesuai dengan perancangan perangkat keras yang sudah ada pada di sub-bab 5.1.2. Sistem menggunakan sensor *Gyroscope*, sensor suara, dan *push button*. Sensor *Gyroscope* berfungsi sebagai mengukur kemiringan sudut manusia saat manusia sedang terjatuh maupun berdiri normal dari sudut *y*, *p*, dan *r*. Sensor suara digunakan sebagai mengukur level *volume* suara manusia saat pengguna sedang berteriak saat keadaan darurat atau sedang berbicara normal, dan *push button* digunakan sebagai *trigger* untuk memulai penyimpanan data yang akan diolah sebelum dikirim ke aplikasi *mobile web*. Berikut adalah rangkaian dari implementasi perangkat keras pada Gambar 5.15 dan 5.16.



Gambar 5.15 Implementasi Rangkaian Perangkat Keras Tampak Samping



Gambar 5.16 Implementasi Rangkaian Perangkat Keras Bagian Dalam

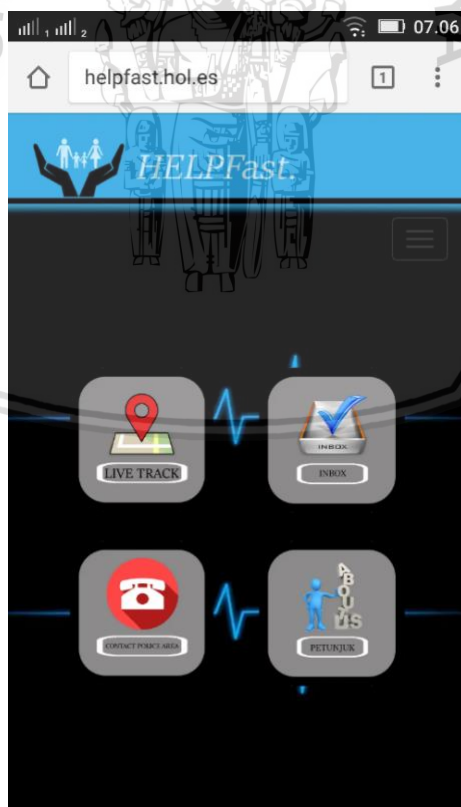
Pada Gambar 5.15 dan Gambar 5.16 yaitu implementasi rangkaian perangkat keras pada sistem yang dibuat dari akrilik berbentuk persegi panjang, lalu menggabungkan komponen-komponen seperti sensor dan modul menjadi satu bagian dengan desain utama. Terdapat juga *push button* yang berada diatas desain utama yang berguna agar memudahkan pengguna untuk menggunakannya secara cepat.

5.2.3 Implementasi Perangkat Lunak

Pada Implementasi ini sub-bab perangkat lunak terdapat beberapa bagian yaitu program pembacaan nilai pada sensor *Gyroscope* dan sensor suara, tampilan *output* pada aplikasi *mobile web*, pembentukan fuzzifikasi, pembentukan rule dari hasil yang didapatkan pada setiap himpunan variabel, inferensi dan juga pembentukan defuzzifikasi untuk mendapatkan hasil *output* yang berada pada sistem. Program akan menggunakan Bahasa C pada Arduino IDE dan program HTML pada Sublime Text.

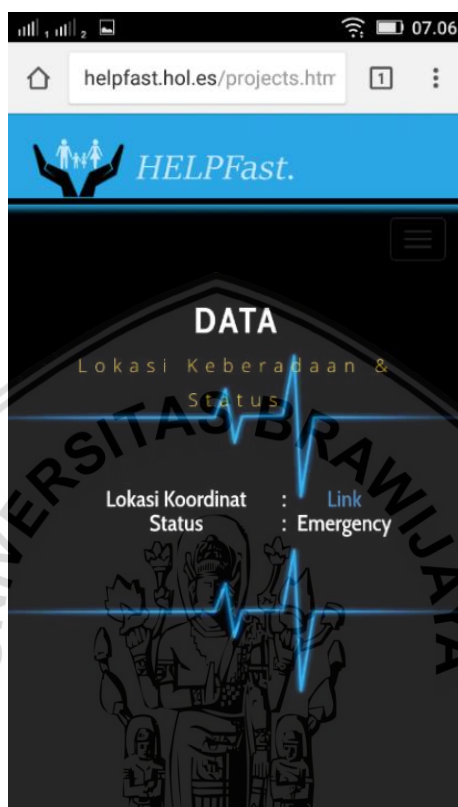
5.2.3.1 Implementasi Aplikasi *Mobile Web*

Dibagian tahap implementasi disini, aplikasi *mobile web* yang dimaksud yaitu aplikasi yang dibuat menggunakan bahasa html melalui sublime text kemudian diupload melalui hosting penulis, ini berguna agar data akurasi pengiriman akan lebih cepat dan tepat untuk mendapatkan data dari Arduino dan kemudian hasil aplikasi yang sudah jadi akan diconvert menjadi aplikasi *mobile web*. Proses pengiriman *output* notifikasi dari sistem ke aplikasi *mobile web* yaitu menggunakan modul ESP8266 dengan penerimaan data menggunakan method post melalui hosting, didalam hosting akan memproses untuk menampilkan pesan berupa emergency pada menu inbox.



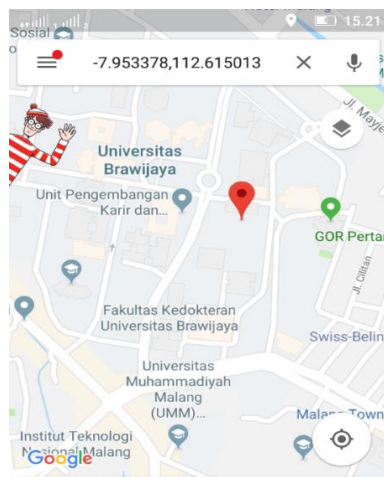
Gambar 5.17 Menu Aplikasi *Mobile Web*

Gambar 5.17 merupakan tampilan menu pada aplikasi *mobile web*. Pada tampilan tersebut terdapat 3 menu utama dan 1 menu tambahan yaitu menu *Live Track* yang berguna untuk mengawasi setiap saat posisi pengguna. 3 menu utama yaitu *inbox*, *contact police area*, dan petunjuk. Masing-masing akan dijelaskan pada keterangan dibawah ini.



Gambar 5.18 Aplikasi *Mobile Web* pada Menu *Inbox*

Pada Gambar 5.18 merupakan tampilan aplikasi *mobile web* yang berada di menu *inbox*. Menu *inbox* berfungsi menerima pesan dari *output* alat yang akan menampilkan status dan lokasi keberadaan *user* saat mengalami keadaan darurat/*emergency*. Untuk lokasi koordinat aplikasi *mobile web* ini membutuhkan aplikasi tambahan berupa google map untuk mencari titik koordinat yang sudah didapatkan melalui modul GPS.

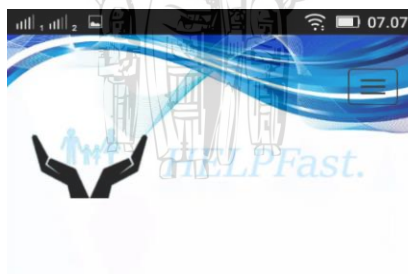


Pin Dipasang



Gambar 5.19 Tampilan pada Google Map

Pada Gambar 5.19 merupakan tampilan setelah menekan link yang terdapat pada menu *inbox*. Diketahui jika posisi petunjuk merah yang terdapat pada gambar merupakan lokasi dimana *user* mengalami kejadian *emergency*. Penulis melakukan percobaan sistem yang sudah jadi di lingkungan Fakultas Ilmu Komputer. Titik koordinat dari modul GPS yang didapatkan adalah -7.953378, 112.615013.

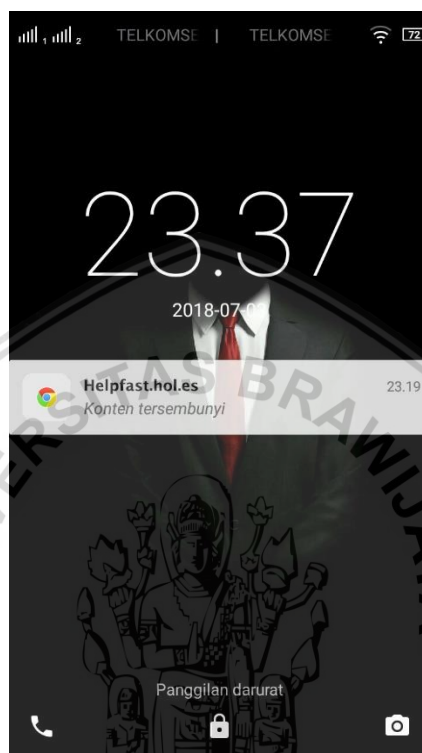


CONTACT POLICE



Gambar 5.20 Aplikasi *Mobile Web* pada Menu *Contact Police Area*

Pada Gambar 5.20 merupakan tampilan aplikasi *mobile web* yang berada di menu *contact police area*. Menu ini berfungsi dimana saat pengguna mengalami kejadian darurat yang jauh dari pihak keluarga dan pengguna membutuhkan pertolongan cepat maka pihak keluarga bisa menghubungi pihak berwajib yang terdekat dari kejadian untuk memastikan apakah posisi koordinat pengguna sistem mengalami keadaan yang darurat.



Gambar 5.21 Aplikasi *Mobile Web* pada Tampilan Notifikasi

Gambar 5.21 merupakan tampilan notifikasi berupa pesan pada layar utama telepon genggam saat user mengalami keadaan darurat. Notifikasi ini bertujuan agar user keluarga terdekat sebagai pihak kedua bisa langsung melihat kondisi user serta mengetahui keberadaan lokasi user di dalam aplikasi *mobile web* dengan cepat.

5.2.3.2 Implementasi Fuzzy Sugeno Pada Alat

Cara mengimplementasikan *fuzzy sugeno*, yaitu yang pertama harus dilakukan untuk memprogram Arduino pada Arduino IDE dengan menginisialisasikan libray apa saja yang digunakan, menginisialisasikan pin yang digunakan dari sensor ke Arduino, dan memvoid setup. Dalam menerapkan fungsi logic ke dalam sistem alat yaitu yang pertama dilakukan sistem dalam membaca *input* dari sensor *Gyroscope*, sensor suara dan *push button*. Dari hasil tersebut akan dibuat fuzzifikasi pada variabel *Gyroscope* dan suara. Program fuzzifikasi di variabel *Gyroscope* seperti Gambar 5.22.

```

1 //=====fuzzifikasi Suara=====
2 //suara rendah
3 If (suara1 <= 16) {
4     suara [0] = 1;
5 } else if (suara1 > 16 && suara1 <= 19){
6     suara[0] = (19 - suara1) / (19 - 16);
7 } else if (suara1 >= 19) {
8     suara[0] = 0;
9 }
10
11
12 //suara sedang
13 If (suara1 <= 27 && suara1 > 19) {
14     suara[1] = 1;
15 } else if (suara1 > 27 && suara1 <= 35) {
16     suara[1] = (35 - suara1) / (35 - 27);
17 } else if (suara1 > 16 && suara1 <= 19) {
18     suara[1] = (suara1 - 16) / (19 - 16);
19 } else if (suara1 >= 35) {
20     suara[1] = 0;
21 }
22
23
24 //suara tinggi
25 If (suara1 <= 200 && suara1 > 35) {
26     suara[2] = 1;
27 } else if (suara1 > 200 && suara1 <= 352) {
28     suara[2] = (352 - suara1) / (352 - 200);
29 } else if (suara1 > 27 && suara1 <= 35) {
30     suara[2] = (suara1 - 27) / (35 - 27);
31 } else if (suara1 >= 352) {
32     suara[2] = 0;
33 }
34
35
36 //suara sangat tinggi
37 If (suara1 >= 352) {
38     suara[3] = 1;
39 } else if (suara1 > 200 && suara1 <= 352) {
40     suara[3] = (suara1 - 200) / (325 - 200);
41 } else if (suara1 <= 200) {
42     suara[3] = 0;
43 }

```

Gambar 5.22 Kode Program Fuzzifikasi Suara

Pada Gambar 5.22 pembacaan dari sensor suara dilakukan untuk mendapatkan himpunan variabel suara yaitu rendah, sedang, tinggi, dan sangat tinggi menggunakan kode yang sama, tetapi untuk mengubah variabel array dan nilai agar sesuai dengan *membership function*. Setelah sensor suara mendapatkan nilai data *input* maka akan diubah menjadi variabel suara yaitu rendah, sedang, tinggi, dan sangat tinggi. Setelah nilai *fuzzy* suara didapatkan maka akan disimpan dalam array suara[4]. Rendah disimpan pada array suara[0], sedang disimpan pada array suara[1], tinggi akan disimpan pada array suara[2], dan sangat tinggi akan disimpan pada array suara[3]. Saat fuzzifikasi suara selesai dibuat, selanjutnya adalah membuat fuzzifikasi untuk sensor *Gyroscope* yang seperti pada Gambar 5.23.

```

1 //=====fuzzifikasi Gyroscope=====
2 //Gyro Y berdiri
3 if (y1 >= 0 && y1 <= 16) {
4     gy_y[0] = 1;
5 } else if (y1 > -62 && y1 <= 0) {
6     gy_y[0] = (y1 - (-62)) / (0 - (-62));
7 } else if (y1 > 16 && y1 <= 65) {
8     gy_y[0] = (65 - y1) / (65 - 16);
9 } else if (y1 >= 65) {
10    gy_y[0] = 0;
11 }
12
13 //Gyro Y jatuh
14 if (y1 > -118 && y1 <= -62 ) {
15     gy_y[1] = 1;
16 } else if (y1 > -62 && y1 <= 0 ) {
17     gy_y[1] = (0 - y1) / (0 - (-62));
18 } else if (y1 >= 0) {
19     gy_y[1] = 0;
20 } else if (y1 >= 65) {
21     gy_y[1] = 1;
22 } else if (y1 > 16 && y1 <= 65) {
23     gy_y[1] = (y1 - 16) / (65 - 16);
24 } else if (y1 < 16) {
25     gy_y[1] = 0;
26 }
27
28 //Gyro P berdiri
29 if (p1 > 0 && p1 <= 2) {
30     gy_p[0] = 1;
31 } else if (p1 > -62 && p1 <= 0) {
32     gy_p[0] = (p1 - (-62)) / (0 - (-62));
33 } else if (p1 > 2 && p1 <= 46) {
34     gy_p[0] = (46 - p1) / (46 - 2);
35 } else if (p1 >= 46) {
36     gy_p[0] = 0;
37 }
38
39
40
41

```

```

42 //Gyro P jatuh
43 if (p1 > -86 && p1 <= -62) {
44     gy_p[1] = 1;
45 } else if (p1 > -62 && y <= 0) {
46     gy_p[1] = (0 - y) / (0 - (-62));
47 } else if (p1 >= 0) {
48     gy_p[1] = 0;
49 } else if (p1 >= 46) {
50     gy_p[1] = 1;
51 } else if (p1 > 2 && p1 <= 46) {
52     gy_p[1] = (p1 - 2) / (46 - 2);
53 } else if (p1 > 2) {
54     gy_p[1] = 0;
55 }
56
57
58 //Gyro R berdiri
59 if (r1 >= 29) {
60     gy_r[0] = 1;
61 } else if (r1 > 20 && r1 <= 29) {
62     gy_r[0] = (r1 - 20) / (29 - 20);
63 } else if (r1 <= 20) {
64     gy_r[0] = 0;
65 }
66
67
68 //Gyro R jatuh
69 if (r1 > -20 && r1 <= 20) {
70     gy_r[1] = 1;
71 } else if (r1 > 20 && r1 <= 29) {
72     gy_r[1] = (29 - x) / (29 - 20);
73 } else if (r1 >= 29) {
74     gy_r[1] = 0;
75 }

```

Gambar 5.23 Kode Program Fuzzifikasi *Sensor Gyroscope*

Gambar 5.23 pembacaan sensor *Gyroscope* dilakukan untuk mendapatkan variabel himpunan Gyro, yaitu *gy_y* terjatuh dan berdiri, *gy_p* terjatuh dan berdiri, dan *gy_r* terjatuh dan berdiri. Setelah mendapatkan nilai *fuzzy Gyroscope* kemudian disimpan dalam array *gy[3]*. *Gyro_y* akan disimpan di array *gy_y[0]* untuk kondisi berdiri dan *gy_y[1]* saat kondisi terjatuh, *Gyro_p* akan disimpan di array *gy_p[0]* saat kondisi berdiri dan *gy_p[1]* saat kondisi terjatuh, *Gyro_r* akan disimpan di array *gy_r[0]* saat kondisi berdiri dan *gy_r[1]* saat keadaan terjatuh.

Setelah pembuatan fuzzifikasi untuk sensor suara dan *Gyroscope*, selanjutnya yaitu memprogram pembuatan rule *fuzzy* dan inferensi. Inferensi dilakukan dengan cara membandingkan nilai array pada variabel suara dan *Gyroscope* dengan fungsi MIN. Berikut Gambar 5.24 merupakan kode program rule *fuzzy* dan inferensi.


```

1 //=====pembuatan rule dan
2 inferensi=====//
3     rule0 = min (suara[0], min(gy_y[0], min( gy_p[0],
4 gy_r[0]))); //normal
5     rule1 = min (suara[0], min(gy_y[0], min( gy_p[0],
6 gy_r[1]))); //normal
7     rule2 = min (suara[0], min(gy_y[0], min( gy_p[1],
8 gy_r[0]))); //normal
9     rule3 = min (suara[0], min(gy_y[0], min( gy_p[1],
10 gy_r[1]))); //normal
11     rule4 = min (suara[0], min(gy_y[1], min( gy_p[0],
12 gy_r[0]))); //normal
13     rule5 = min (suara[0], min(gy_y[1], min( gy_p[0],
14 gy_r[1]))); //normal
15     rule6 = min (suara[0], min(gy_y[1], min( gy_p[1],
16 gy_r[0]))); //normal
17     rule7 = min (suara[0], min(gy_y[1], min( gy_p[1],
18 gy_r[1]))); //emergency
19     rule8 = min (suara[1], min(gy_y[0], min( gy_p[0],
20 gy_r[0]))); //normal
21     rule9 = min (suara[1], min(gy_y[0], min( gy_p[0],
22 gy_r[1]))); //normal
23     rule10 = min (suara[1], min(gy_y[0], min( gy_p[1],
24 gy_r[0]))); //normal
25     rule11 = min (suara[1], min(gy_y[0], min( gy_p[1],
26 gy_r[1]))); //normal
27     rule12 = min (suara[1], min(gy_y[1], min( gy_p[0],
28 gy_r[0]))); //normal
29     rule13 = min (suara[1], min(gy_y[1], min( gy_p[0],
30 gy_r[1]))); //normal
31     rule14 = min (suara[1], min(gy_y[1], min( gy_p[1],
32 gy_r[0]))); //normal
33     rule15 = min (suara[1], min(gy_y[1], min( gy_p[1],
34 gy_r[1]))); //emergency
35     rule16 = min (suara[2], min(gy_y[0], min( gy_p[0],
36 gy_r[1]))); //emergency
37     rule17 = min (suara[2], min(gy_y[0], min( gy_p[0],
38 gy_r[1]))); //emergency
39     rule18 = min (suara[2], min(gy_y[0], min( gy_p[1],
40 gy_r[0]))); //emergency
41     rule19 = min (suara[2], min(gy_y[0], min( gy_p[1],
42 gy_r[1]))); //emergency
43     rule20 = min (suara[2], min(gy_y[1], min( gy_p[0],
44 gy_r[0]))); //emergency
45     rule21 = min (suara[2], min(gy_y[1], min( gy_p[0],
46 gy_r[1]))); //emergency
47     rule22 = min (suara[2], min(gy_y[1], min( gy_p[1],
48 gy_r[0]))); //emergency
49     rule23 = min (suara[2], min(gy_y[1], min( gy_p[1],
50 gy_r[1]))); //emergency
51     rule24 = min (suara[3], min(gy_y[0], min( gy_p[0],
52 gy_r[0]))); //emergency
53     rule25 = min (suara[3], min(gy_y[0], min( gy_p[0],
54 gy_r[1]))); //emergency
55     rule26 = min (suara[3], min(gy_y[0], min( gy_p[1],
56 gy_r[0]))); //emergency
57     rule27 = min (suara[3], min(gy_y[0], min( gy_p[1],
58 gy_r[1]))); //emergency

```

31	rule28 = min (suara[3], min(gy_y[1], min(gy_p[0], gy_r[0]))); //emergency
32	rule29 = min (suara[3], min(gy_y[1], min(gy_p[0], gy_r[1]))); //emergency
33	rule30 = min (suara[3], min(gy_y[1], min(gy_p[1], gy_r[0]))); //emergency
	rule31 = min (suara[3], min(gy_y[1], min(gy_p[1], gy_r[1]))); //emergency

Gambar 5.24 Kode Program Rule dan Inferensi

Didalam Gambar 5.24 merupakan rule *fuzzy* yang dibuat berdasarkan dari kombinasi variabel himpunan suara dan *Gyroscope*. Terdapat 32 rule dari kombinasi variabel himpunan suara dan *Gyroscope*. Pembuat rule menggunakan fungsi implikasi. Didalam metode sugeno, fungsi implikasi yang digunakan adalah MIN. saat mengambil salah satu rule yang diambil sebagai inferensi akan menggunakan fungsi MAX.

Setelah proses inferensi, selanjutnya yaitu fungsi defuzzifikasi. Defuzzifikasi merupakan proses terakhir *fuzzy* sugeno. Didalam *fuzzy* sugeno, fungsi yang digunakan untuk defuzzifikasi yaitu MAX. Berikut Gambar 5.25 merupakan kode program defuzzifikasi.

1	//=====defuzzifikasi=====//
2	defu1 = max(rule0, max(rule1, max(rule2, rule3)));
3	defu2 = max(rule4, max(rule5, max(rule6, rule7)));
4	defu3 = max(rule8, max(rule9, max(rule10, rule11)));
5	defu4 = max(rule12, max(rule13, max(rule14, rule15)));
6	defu5 = max(rule16, max(rule17, max(rule18, rule19)));
7	defu6 = max(rule20, max(rule21, max(rule22, rule23)));
8	defu7 = max(rule24, max(rule25, max(rule26, rule27)));
9	defu8 = max(rule28, max(rule29, max(rule30, rule31)));
10	defu_final1 = max(defu1, max(defu2, defu3));
11	defu_final2 = max(defu4, max(defu5, defu6));
12	defu_final3 = max(defu7, defu8);
13	defu_final= max(defu_final1, max(defu_final2, defu_final3));

Gambar 5.25 Kode Program Defuzzifikasi

Gambar 5.25 merupakan potongan program defuzzifikasi menggunakan fungsi MAX. Proses dilakukan menggunakan cara perbandingan hasil pada setiap *output* yang telah diproses inferensi seperti Gambar 5.26. Ini bertujuan sebagai perbandingan hasil di masing-masing *output* yaitu untuk mendapatkan nilai terbesar yang hasilnya disimpan di variabel *output*. Setelah mendapatkan hasil proses defuzzifikasi kemudian dimasukkan ke dalam proses if-else yang fungsinya sebagai pengeluaran *output* yang akan dikirim ke aplikasi *mobile web* jika kondisi pengguna sedang terjadi keadaan darurat. Berikut Gambar 5.26 ini adalah kode program untuk pengeluaran *output* yang sebagai penentu pengiriman ke aplikasi *mobile web*.

```

1      if(defu_final==rule0){
2          return "Normal";//ini hasil output rule0
3      }else if(defu_final==rule1){
4          return "Normal";//ini hasil output rule1
5      }else if(defu_final==rule2){
6          return "Normal";//ini hasil output rule2
7      }else if(defu_final==rule3){
8          return "Normal";//ini hasil output rule3
9      }else if(defu_final==rule4){
10         return "Normal";//ini hasil output rule4
11     }else if(defu_final==rule5){
12         return "Normal";//ini hasil output rule5
13     }else if(defu_final==rule6){
14         return "Normal";//ini hasil output rule6
15     }else if(defu_final==rule7){
16         return "Emergency";//ini hasil output rule7
17     }else if(defu_final==rule8){
18         return "Normal";//ini hasil output rule8
19     }else if(defu_final==rule9){
20         return "Normal";//ini hasil output rule9
21     }else if(defu_final==rule10){
22         return "Normal";//ini hasil output rule10
23     }else if(defu_final==rule11){
24         return "Normal";//ini hasil output rule11
25     }else if(defu_final==rule12){
26         return "Normal";//ini hasil output rule12
27     }else if(defu_final==rule13){
28         return "Normal";//ini hasil output rule13
29     }else if(defu_final==rule14){
30         return "Normal";//ini hasil output rule14
31     }else if(defu_final==rule15){
32         return "Emergency";//ini hasil output rule15
33     }else if(defu_final==rule16){
34         return "Normal";//ini hasil output rule16
35     }else if(defu_final==rule17){
36         return "Emergency";//ini hasil output rule17
37     }else if(defu_final==rule18){
38         return "Emergency";//ini hasil output rule18
39     }else if(defu_final==rule19){
40         return "Emergency";//ini hasil output rule19
41     }else if(defu_final==rule20){
42         return "Emergency";//ini hasil output rule20
43     }else if(defu_final==rule21){
44         return "Emergency";//ini hasil output rule21
45     }else if(defu_final==rule22){
46         return "Emergency";//ini hasil output rule22
47     }else if(defu_final==rule23){
48         return "Emergency";//ini hasil output rule23
49     }else if(defu_final==rule24){
50         return "Emergency";//ini hasil output rule24
51     }else if(defu_final==rule25){
52         return "Emergency";//ini hasil output rule25
53     }else if(defu_final==rule26){
54         return "Emergency";//ini hasil output rule26
55     }else if(defu_final==rule27){
56         return "Emergency";//ini hasil output rule27
57     }else if(defu_final==rule28){

```

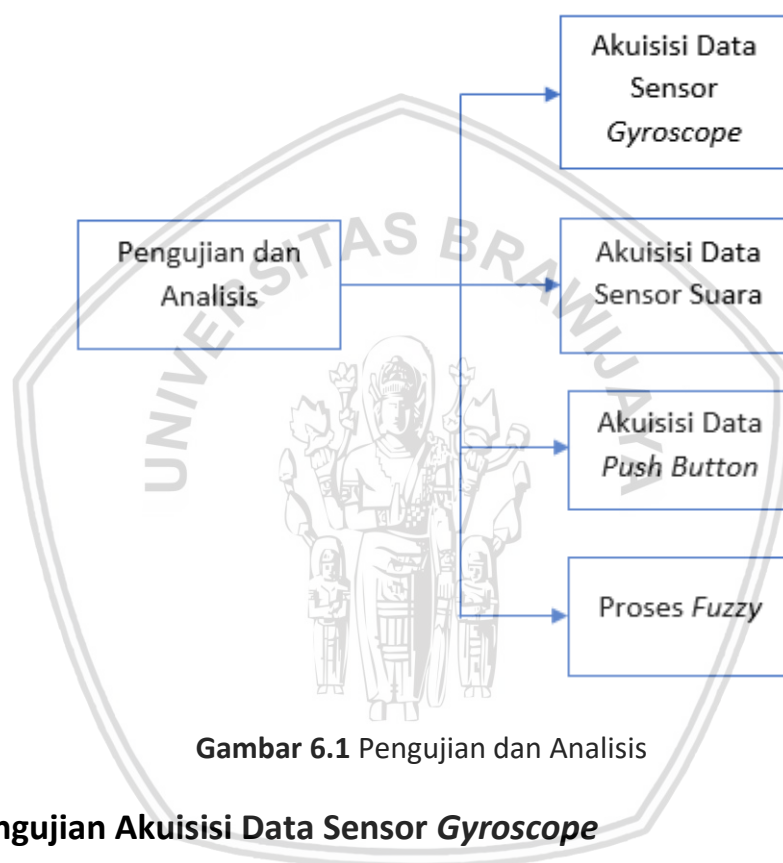
58	return "Emergency";//ini hasil output rule28
59	}else if(defu_final==rule29){
60	return "Emergency";//ini hasil output rule29
61	}else if(defu_final==rule30){
62	return "Emergency";//ini hasil output rule30
63	}else if(defu_final==rule31){
64	return "Emergency";//ini hasil output rule31
65	}

Gambar 5.26 Kode Program *Output* Sistem



BAB 6 PENGUJIAN DAN ANALISIS

Didalam bab pengujian dan analisis akan membahas mengenai proses pengujian pada penelitian yaitu “Implementasi Sistem Notifikasi Keadaan Darurat Berbasis Aplikasi *Mobile Web* dan Arduino Mega Menggunakan Logika *Fuzzy*”. Ada 4 pengujian yang akan dilakukan, yaitu pengujian tentang akuisisi data sensor *Gyroscope*, akuisisi data sensor suara, akuisisi data *push button* dan proses *fuzzy*. Pengujian dijelaskan dalam Gambar 6.1.



Gambar 6.1 Pengujian dan Analisis

6.1 Pengujian Akuisisi Data Sensor *Gyroscope*

1. Tujuan

Tujuan dalam melakukannya pengujian akuisisi data pada sensor *Gyroscope* yaitu untuk mengetahui kemiringan dari sudut *Gyroscope* ketika kemiringan menandakan bahwa *Gyroscope* masih normal atau terjatuh. Pengujian dilakukan dengan 2 perlakuan kepada sensor *Gyroscope* untuk menentukan kondisi sudut tegak normal ataupun kondisi *Gyroscope* terjatuh.

2. Prosedur

Prosedur yang ada dan harus dilakukan untuk melakukan pengujian sensor *Gyroscope* yaitu :

1. Merancang sensor *Gyroscope* dengan Arduino Mega 2560 menggunakan kabel jumper, dengan menyatukan sensor dengan kontroler melalui pin-pinya.
2. Membuka Arduino IDE kemudian memprogram kodingan untuk membaca sensor *Gyroscope*.
3. Mengcompile kodingan dan kemudian mengupload kode program yang telah dibuat dan benar.
4. Menekan *button* sebagai *trigger* selama 7 detik dan memberikan masukan *input* pada sensor *Gyroscope* dengan 2 kondisi pengguna berdiri normal ataupun terjatuh.
5. Mengamati hasil *output* pada sensor *Gyroscope* yang berada pada serial monitor Arduino IDE, dan mengambil 20 sampel dari hasil pengujian.
6. Kesimpulan.

3. Hasil dan Analisis

Berikut ini adalah Tabel 6.1 yaitu hasil dari pembacaan sensor *Gyroscope* dari masing-masing sudut Y, P, dan R setelah melalui prosedur sehingga didapatkannya dari sensor *Gyroscope*.

Tabel 6.1 Hasil Pengujian Sensor *Gyroscope*

Percobaan ke -	Sudut Gyro Y	Sudut Gyro P	Sudut Gyro R
1	-25.49	-27.5	57.75
2	-98.31	-85.6	3.75
3	-74.98	0.57	-0.6
4	-160.46	2.21	-10.72
5	0.16	0.23	87.57
6	-51.85	0.99	-11.59
7	-1.75	1.02	-11.68
8	2.74	-0.43	-1.6
9	-0.08	-2.55	83.7
10	-25.3	0.78	1.75

Percobaan ke -	Sudut Gyro Y	Sudut Gyro P	Sudut Gyro R
11	-10.78	-0.55	77.33
12	1.74	-2.05	85.55
13	-70.85	63.91	22.98
14	8.98	-3.58	84.17
15	8.12	-7.55	-8.72
16	18.88	-7.78	78.35
17	0.62	2.52	86.59
18	-17.42	-1.14	-7.97
19	15.8	-3.08	80.32
20	-9.73	11.05	74.6

Tabel 6.1 merupakan hasil pembacaan dari sensor *Gyroscope* setelah melalui berbagai prosedur sehingga didapatkannya data dari sensor *Gyroscope*. Untuk hasil kondisi akhir *output* terjatuh maupun berdiri normal terdapat pada Tabel 6.4 ini dikarenakan pengujian sistem tergabung dengan sensor suara dan sudah melalui proses logika *fuzzy*, yang dimana logika *fuzzy* akan menentukan hasil *output* melalui 2 kesimpulan nilai data yang diambil melalui sensor *Gyroscope* dan sensor suara.

6.2 Pengujian Akuisisi Data Sensor Suara

1. Tujuan

Tujuan dari dilakukannya pengujian akuisisi data pada sensor suara yaitu untuk mengetahui level *volume* suara melalui *input* suara manusia. Level *volume* suara manusia itu nantinya akan digunakan sebagai *input* serta menentukan *output* yang akan dioalah melalui logika *fuzzy*.

2. Prosedur

Prosedur yang ada dan harus dilakukan untuk melakukan pengujian sensor Suara yaitu :

1. Merancang sensor suara dengan Arduino Mega 2560 menggunakan kabel jumper, dengan menyatukan sensor dengan kontroler melalui pin-pinya.
2. Membuka Arduino IDE kemudian memprogram kodingan untuk membaca nilai level *volume* pada sensor suara.
3. Mengcompile kodingan dan kemudian mengupload kode program yang telah dibuat dan benar.
4. Menekan *button* sebagai *trigger* selama 7 detik dan memberikan masukan *input* pada sensor suara untuk menentukan 2 kondisi pengguna yaitu berteriak membutuhkan bantuan ataupun berbicara dengan normal.
5. Mengamati hasil *output* pada sensor suara yang berada pada serial monitor Arduino IDE, dan mengambil 20 sampel dari hasil pengujian.
6. Kesimpulan.

3. Hasil dan Analisis

Berikut ini adalah Tabel 6.2 yaitu hasil dari pembacaan sensor suara pada sistem dan setelah melalui prosedur sehingga didapatkannya dari sensor suara.

Tabel 6.2 Hasil Pengujian Sensor Suara

Percobaan Ke-	Pembacaan Sensor Suara
1	23
2	44
3	35
4	19
5	19
6	83
7	21
8	47
9	469
10	52

Percobaan Ke-	Pembacaan Sensor Suara
11	838
12	33
13	33
14	47
15	47
16	48
17	328
18	21
19	151
20	290

Tabel 6.2 merupakan hasil pembacaan dari sensor suara setelah melalui berbagai prosedur sehingga didapatkannya nilai data dari sensor suara. Untuk hasil kondisi akhir *output* berteriak membutuhkan bantuan ataupun berbicara dengan normal terdapat pada Tabel 6.4 ini dikarenakan pengujian sistem tergabung dengan sensor *Gyroscope* dan sudah melalui proses logika *fuzzy*, yang dimana logika *fuzzy* akan menentukan hasil *output* melalui 2 kesimpulan nilai data yang diambil melalui sensor *Gyroscope* dan sensor suara.

6.3 Pengujian Akuisisi Data Push Button

1. Tujuan

Tujuan melakukan pengujian akuisisi data pada *push button* yaitu untuk mengetahui berfungsinya *push button* sebagai *trigger* atau tidak dan mengambil kesimpulan nilai data dari sensor suara dan *Gyroscope*.

2. Prosedur

Prosedur yang ada dan harus dilakukan untuk melakukan pengujian akuisisi data pada *push button* yaitu :

1. Merancang *push button* dengan rangkaian debounce kemudian disambungkan Arduino Mega 2560 menggunakan kabel jumper melalui pin-pinya.
2. Membuka Arduino IDE kemudian memprogram kodingan untuk membaca nilai pada *push button*.
3. Mengcompile kodingan dan kemudian mengupload kode program yang telah dibuat dan benar.
4. Menekan *button* sebagai *trigger* selama 7 detik dan mengamati apakah *push button* bernilai 1 atau 0..
5. Kesimpulan.

3. Hasil dan Analisis

Berikut ini adalah Tabel 6.3 yaitu hasil dari pembacaan *trigger* pada *push button*.

Tabel 6.3 Hasil Pengujian *Push Button*

Percobaan Ke-	Kondisi <i>Push Button</i>	Pembacaan <i>Trigger Push Button</i>
1	Tertekan	1
2	Tertekan	1
3	Tertekan	1
4	Tidak Tertekan	0
5	Tidak Tertekan	0
6	Tidak Tertekan	0

Di dalam Tabel 6.3 merupakan hasil dari pengujian pembacaan dari *push button* setelah melalui berbagai prosedur sehingga didapatkan nilai data dari *push button*. Dari tabel menunjukkan bahwa ketika *push button* ditekan, *trigger* pengambilan keputusan nilai dari sensor *Gyroscope* dan sensor suara data yang didapat adalah 1. Dan ketika *push button* tidak ditekan sebagai *trigger* maka data yang didapat adalah 0 yaitu sensor suara dan *Gyroscope* tidak dapat disimpulkan untuk mendapatkan *input* data.

6.4 Pengujian Proses Fuzzy

1. Tujuan

Tujuan melakukan pengujian pada proses *fuzzy* yaitu untuk mengetahui keberhasilan sistem agar dapat menentukan keputusan atau kesimpulan dari sistem yang akan dikirim ke aplikasi *mobile web*.

2. Prosedur

Prosedur yang ada dan harus dilakukan untuk melakukan pengujian proses *fuzzy* yaitu :

1. Merancang sensor suara, *push button*, sensor suara, modul GPS, dan modul ESP8266 dengan Arduino Mega 2560 menggunakan kabel jumper, dengan menyatukan sensor dan modul dengan kontroler melalui pin-pinya.
2. Membuka Arduino IDE kemudian memprogram kodingan logika *fuzzy*.
3. Mengcompile kodingan dan kemudian mengupload kode program yang telah dibuat.
4. Buka serial monitor yang terdapat di Arduino Mega untuk mengamati hasil *output fuzzy*.
5. Pastikan modul wifi sudah connect dengan jaringan yang ada, jika sudah connect selanjutnya tekan *push button* sebagai *trigger* selama 7 detik dan berikan masukan *input* pada masing-masing sensor.
6. Mengamati hasil yang ditampilkan di serial monitor Arduino IDE, dan mengambil 20 sampel dari hasil pengujian. Hasil *output* mempresentasikan 2 kondisi yaitu *emergency* dan normal.
7. Melakukan perhitungan logika *fuzzy* dari hasil pembacaan sensor *Gyroscope* dan sensor suara secara manual. Kemudian membandingkan dengan hasil *output* yang didapatkan pada serial monitor.
8. Kesimpulan.

3. Hasil dan Analisis

Pada Tabel 6.4 menggunakan 20 sampel yang sudah didapatkan dengan *input* dan *output* yaitu *emergency* dan normal. Pengujian dilakukan dengan cara menganalisis dari rumus *fuzzy* yang telah diterapkan pada sistem yang kemudian memberikan keputusan atau kesimpulan pada *output* yang sesuai dengan perhitungan *fuzzy*. Berikut adalah pengujian yang menggunakan perhitungan manual berdasarkan perhitungan *fuzzy*. Sampel yang akan diambil yaitu pada percobaan 13 yang terdapat pada tabel 6.4.

- Suara : 33
- Gyro_Y : -70.85
- Gyro_P : 63.91
- Gyro_R : 22.98

1. Fuzzifikasi

➤ Suara

Rendah : 0

$$\text{Sedang} : \frac{35 - 33}{35 - 27} = 0.25$$

$$\text{Tinggi} : \frac{33 - 27}{35 - 27} = 0.75$$

Sangat Tinggi : 0

➤ Gyro_Y

Berdiri : 0

Jatuh : 1

➤ Gyro_P

Berdiri : 0

Jatuh : 1

➤ Gyro_R

$$\text{Berdiri} : \frac{22.98 - 20}{29 - 20} = 0.331$$

$$\text{Jatuh} : \frac{29 - 22.98}{29 - 20} = 0.668$$

2. Inferensi

Rule[14] : IF SUARA SEDANG && GYRO Y TERJATUH && GYRO P TERJATUH && GYRO R BERDIRI

$$\begin{aligned} \alpha\text{-predikat1} &= \min (\mu_{\text{SuaraSEDANG}} \cap \mu_{\text{YJATUH}} \cap \mu_{\text{PJATUH}} \cap \mu_{\text{RBERDIRI}}) \\ &= \min (0.25 \cap 1 \cap 1 \cap 0.331) \\ &= 0.25 \end{aligned}$$

Rule[15] : IF SUARA SEDANG && GYRO Y TERJATUH && GYRO P TERJATUH && GYRO R TERJATUH

$$\begin{aligned} \alpha\text{-predikat1} &= \min (\mu_{\text{SuaraSEDANG}} \cap \mu_{\text{YJATUH}} \cap \mu_{\text{PJATUH}} \cap \mu_{\text{RJATUH}}) \\ &= \min (0.25 \cap 1 \cap 1 \cap 0.668) \\ &= 0.25 \end{aligned}$$

Rule[22] : IF SUARA TINGGI && GYRO Y JATUH && GYRO P JATUH && GYRO R BERDIRI

$$\begin{aligned}\alpha\text{-predikat1} &= \min (\mu_{\text{SuaraTINGGI}} \cap \mu_{\text{YJATUH}} \cap \mu_{\text{PJATUH}} \cap \mu_{\text{RBERDIRI}}) \\ &= \min (0.75 \cap 1 \cap 1 \cap 0.331) \\ &= 0.331\end{aligned}$$

Rule[23] : IF SUARA TINGGI && GYRO Y JATUH && GYRO P JATUH && GYRO R JATUH

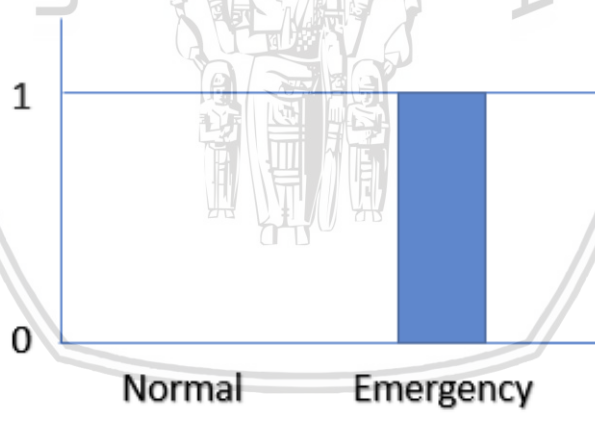
$$\begin{aligned}\alpha\text{-predikat1} &= \min (\mu_{\text{SuaraTINGGI}} \cap \mu_{\text{YJATUH}} \cap \mu_{\text{PJATUH}} \cap \mu_{\text{RJATUH}}) \\ &= \min (0.75 \cap 1 \cap 1 \cap 0.668) \\ &= 0.668\end{aligned}$$

3. Defuzzifikasi

$Output = \max(\text{rule0, rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10, rule11, rule12, rule13, rule14, rule15, rule16, rule 17, rule19, rule20, rule21, rule22, rule23, rule24, rule25, rule26, rule27, rule28, rule29, rule31})$

$Output = \max(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.25, 0.25, 0, 0, 0, 0, 0, 0, 0.331, 0.668, 0, 0, 0, 0, 0, 0, 0, 0)$

$Output = \max(\text{rule23}) = \text{Kondisi Emergency}$



Gambar 6.2 Grafik *Output* Sistem

Didalam Gambar 6.2 menunjukkan bahwa *output* sistem adalah *emergency* dengan nilai himpunannya yaitu 1. *Output* yang dihasilkan dari perhitungan manual sesuai dengan perhitungan sistem. Hal ini menunjukkan bahwa sistem sudah sesuai dengan kebutuhan yaitu mengeluarkan *output* sistem yang sesuai dengan perancangan dan sudah memenuhi kebutuhan yang ada pada fungsional sistem. Berikut Tabel 6.4 merupakan hasil analisis pengujian tingkat keakuratan proses *fuzzy*.

Tabel 6.4 Hasil Pengujian Fuzzy

Percobaan ke-	Pembacaan Sensor Suara	Pembacaan Sensor Gyro_Y	Pembacaan Sensor Gyro_P	Pembacaan Sensor Gyro_R	Output Sistem	Output Secara Manual	Kondisi
1	23	-25.49	-25.5	57.75	0	0	Normal
2	44	-98.31	-85.6	3.75	1	1	<i>Emergency</i>
3	35	-74.98	0.57	-0.6	1	1	<i>Emergency</i>
4	19	-160.46	2.21	-10.72	0	0	Normal
5	19	0.16	0.23	87.57	0	0	Normal
6	83	-51.85	0.99	-11.59	1	1	<i>Emergency</i>
7	21	-1.75	1.02	-11.68	0	0	Normal
8	47	2.75	-0.43	-1.6	1	1	<i>Emergency</i>
9	469	-0.08	-2.55	83.7	1	1	<i>Emergency</i>
10	52	-25.3	0.78	1.75	1	1	<i>Emergency</i>
11	838	-10.78	-0.55	77.33	1	1	<i>Emergency</i>
12	33	1.74	-2.05	85.55	1	1	<i>Emergency</i>
13	33	-70.85	63.91	22.98	1	1	<i>Emergency</i>
14	47	8.98	-3.58	84.17	1	1	<i>Emergency</i>
15	47	8.12	-7.75	-8.72	1	1	<i>Emergency</i>
16	48	18.88	-7.78	78.35	1	1	<i>Emergency</i>
17	328	0.62	2.52	86.59	1	1	<i>Emergency</i>
18	21	-17.42	-1.14	-7.97	0	0	Normal
19	151	15.8	-3.08	80.32	1	1	<i>Emergency</i>
20	290	-9.73	11.05	74.6	1	1	<i>Emergency</i>

BAB 7 PENUTUP

Didalam bab ini menjelaskan kesimpulan dari penelitian “Implementasi Sistem Notifikasi Keadaan Darurat Berbasis Aplikasi *Mobile Web* dan Arduino Mega Menggunakan Logika *Fuzzy*” serta saran sebagai pengembangan topik skripsi penelitian lebih lanjut atau penelitian selanjutnya.

7.1 Kesimpulan

Kesimpulan didapatkan berdasarkan dari hasil pengujian dan juga Analisa yang telah dilakukan terhadap skripsi ini sebagai berikut.

1. Proses pada perancangan sistem notifikasi saat keadaan darurat berbasis aplikasi *mobile web* dengan menyambungkan sensor suara, sensor *Gyroscope*, modul ESP8266, modul GPS, dan *push button* ke mikrokontroler Arduino Mega 2560 menjadi sebuah sistem yang terhubung satu sama lain kemudian mengirimkan *output* berupa notifikasi ke aplikasi *mobile web* saat user mengalami keadaan darurat. Proses akuisisi pada sensor suara meliputi pembacaan suara pada manusia. Dan akuisisi pada sensor *Gyroscope* meliputi pembacaan sudut kemiringan sudut pada manusia. Sedangkan proses akuisisi data pada *push button* untuk pembacaan *trigger* apakah *trigger* berfungsi atau tidak.
2. Proses menampilkan lokasi *user* saat mengalami keadaan darurat dan sesuai dengan proses logika fuzzy, maka sistem akan menginput lokasi terakhir pada titik koordinat yang didapat oleh modul GPS kemudian modul ESP8266 akan mengirim lokasi user dengan method post yang kemudian akan diterima dan diproses untuk ditampilkan didalam hosting penulis. Untuk mengakses lokasi titik koordinat penulis menyambungkan dengan aplikasi google map agar akuisisi data tepat dan akurat.
3. Implementasi logika *fuzzy* segeno didalam sistem notifikasi keadaan darurat berbasis aplikasi *mobile web* meliputi 3 proses yang harus dilakukan yaitu proses fuzzifikasi, proses inferensi, dan proses defuzzifikasi. Proses fuzzifikasi terdapat 2 variabel, yaitu variabel suara dan variabel *Gyroscope*. Didalam variabel suara terdapat 4 himpunan yaitu himpunan rendah, sedang, tinggi, dan sangat tinggi. Sedangkan pada variabel *Gyroscope* terdapat 3 himpunan yaitu himpunan Gyro Y, Gyro P, dan Gyro R. Didalam proses inferensi yang merupakan penggabungan dari seluruh aturan (*rule*) yang didasari dari data himpunan pada setiap variabel yaitu variabel suara dan variabel *Gyroscope*. Terdapat 32 aturan (*rule*) *fuzzy* yang ada pada sistem notifikasi keadaan darurat ini dengan perintah “IF” dan “AND” yang kemudian menghasilkan perintah “THEN”. Di proses terakhir yaitu defuzzifikasi, pada proses ini yaitu menggunakan metode MIN-MAX yang pada tahap selanjutnya di setiap

variable *output* akan dicari melalui nilai terbesarnya (MAX) melalui perhitungan MIN.

4. Pengujian pada logika *fuzzy* dilakukan dengan membandingkan antara perhitungan pada sistem dan perhitungan secara manual. Hasil perhitungan tersebut nantinya akan menunjukkan hasil yang sama. Hal itu menunjukkan bahwa hasil dari pengujian menggunakan metode *fuzzy* pada sistem sudah sesuai dengan perancangan yang dibuat. Sistem notifikasi keadaan darurat ini dapat menentukan 2 kondisi yaitu kondisi *emergency* atau normal yang artinya baik-baik saja dengan *input* suara dan kemiringan manusia yang bervariasi. Didalam pengujian logika *fuzzy*, didapatkan presentasi dari keberhasilan sistem sebesar 100%. Ini menunjukkan bahwa sistem notifikasi keadaan darurat telah berkerja dengan baik dan sesuai dengan kebutuhan dalam menentukan kondisi *user* yang akan ditampilkan melalui aplikasi *mobile web*.

7.2 Saran

Berdasarkan hasil dan pembuatan pada pengujian "Implementasi Sistem Notifikasi Keadaan Darurat Berbasis Aplikasi *Mobile Web* dan Arduino Mega Menggunakan Logika *Fuzzy*", dan setelah menganalisis skripsi ini, diharapkan pada penelitian yang akan dilakukan selanjutnya dapat melakukan pengembangan atau memperbaiki kekurangan yang terdapat dalam sistem yang telah dibangun ini. Saran untuk pengembangan penelitian selanjutnya yaitu :

1. Menggunakan sensor ISD 1820 sebagai modul perekam suara untuk memastikan kejadian yang lebih nyata.
2. Menggunakan sensor getar yang memiliki sensitivitas yang lebih tinggi.
3. Ukuran sistem dapat dibuat lebih kecil dari sebelumnya agar lebih nyaman saat pemakaian.
4. Menggunakan Metode Logika Bayes

DAFTAR PUSTAKA

- Adnan, T. H., & Indriani, S. L. (2016). Aplikasi Tombol Peringatan Darurat Berbasis Android. *Jurnal Sistem Komputer Unikom dan Komputika*.
- Ahmadi, Z. (2013). Keadaan Darurat.
- Ariansyah, J., Heryanto, & Iis, P. (2016). Rancang Bangun Sistem Informasi Tanggap Darurat Divisi Pusat Pengaduan Berbasis Website Di Kota Palembang.
- Ecadio. (2018). Modul GPS Unblox NEO6M. Retrieved from <http://ecadio.com/jual-modul-gps-neo6mv2>
- Henderi, Dian, N. P., & Pamungkas, D. (2016). Rancang Bangun *Automatic Emergency System* Berbasis *Mobile*.
- Naba. (2009). Metode *Fuzzy Sugeno*
- Okazii. (2017, Februari 28). Sensor Suara LM393 Retrieved from <https://www.okazii.ro/modul-senzor-de-sunet-sound-detector-detection-module-arduino-a178638413>
- Playground.arduino. (2016, Maret 25). Sensor *Gyroscope Accelerometer* MPU6050 Retrieved from <https://playground.arduino.cc/Main/MPU-6050>
- Ramadhan, H., Maulana, R. & Hannats, M. H. (2018, September 9). *Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Silhouette* Hewan Menggunakan Logika *Fuzzy*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* vol. 5.
- Rizki, M. S., & Sutanto. (2017). *Prototype Panic Button* Dengan Notifikasi SMS dan Peta Digital.
- Sinauarduino. (2016, April 6). Modul Wifi ESP8266. Retrieved from <http://www.sinauarduino.com/artikel/esp8266/>
- Sparkfun. (2003). *Push Button*. Retrieved from <https://www.sparkfun.com/products/97>
- Sudrajat. (2008). Dasar-dasar *fuzzy*, 30-31
- Technobotsonline. (2018). Arduino Mega 2560. Retrieved from <https://www.technobotsonline.com/arduino-mega-2560-r3.html>